



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

Diseño y desarrollo de una API para la visualización de
datos espaciales en sistemas Web de información
geográfica para dispositivos móviles

Tesis que presenta

Carlos Hernández Nava

para obtener el Grado de

Maestro en Ciencias

en Computación

Director de la Tesis

Dr. Sergio Víctor Chapa Vergara

México, D.F.

Febrero 2009

Resumen

Este trabajo presenta el diseño, desarrollo y validación de una API para la visualización de datos espaciales y su incorporación a sistemas de información geográfica en Web (SIG en Web) en dispositivos móviles. La API debe cumplir con un diseño y desarrollo enfocado a la reutilización de componentes, para la creación de SIG en Web sobre el paradigma Web 2.0. El diseño se realiza con este enfoque por que facilita a los desarrolladores de SIG en Web que no sean expertos, añadir componentes a sus sistemas y realizar consultas geográficas en lenguaje SQL. El sistema gestor de bases de datos espacial es *PostgreSQL* con *PostGIS* para dar soporte a datos espaciales.

Los componentes generados en este trabajo son: una biblioteca de *tags*-JSP para hacer consultas a la base de datos espacial (*GISWEBTAG v. 1.0*); y *scripts* para la generación dinámica de objetos geográficos para su visualización en el navegador Web, en formatos gráficos Canvas y SVG (*Scalable Vector Graphic*). A través de la pruebas y validaciones de la biblioteca se compara la resolución que ofrecen los gráficos Canvas y SVG en dispositivos móviles.

La fuente de los datos espaciales de los SIG desarrollados proviene de INEGI y en específico, son del estado de Colima, México. Los dispositivos móviles para los cuales desarrollo son el *iPodTouch* y *iPhone* de Apple, con su navegador Web Safari en su versión para móviles, de los cuales se aprovecha su pantalla sensible al tacto para realizar acercamientos de forma amigable y rápida. Se desarrollan dos aplicaciones para validar la biblioteca de *tags* JSP, la primera de estas se enfoca al desarrollo sustentable de hidrología, y la segunda para dar soporte a la toma de decisiones en la infraestructura educativa.

Abstract

This work establishes the design, development and testing of a API for visualizing geographic objects in order to incorporate them to WebGIS, a Web Geographic Information System mainly design to support mobile devices. The API design and implementation focus on reusing the developed components in WebGIS over the Web 2.0 paradigm. The design is created to facilitate for inexpert developers of WebGIS, only with add our components could eject spatial queries with SQL language. The spatial database management system is *PostgreSQL* with *PostGIS* to support spatial data.

The components created are: a tags-JSP library (**GISWEBTAG v 1.0**); and scripts to generate dynamically geographic objects in order to visualize them like Canvas or SVG (Scalable Vector Graphic) graphic format on Web browsers. Through the testings of the library we compare the resolution that provide the graphic formats Canvas and SVG.

The source of spatial data of the GIS to validate the API is INEGI, specifically, the Colima state, Mexico. The GIS were created for mobile devices of Apple, iPodTouch and iPhone that provide a multi touch screen that facilitate the zoom in friendly and quickly. The GIS developed were: a Geographic Information System to help the sustainable development of hidrology and a Geographic Information System to support the taking of decision in the education infrastructure.

Agradecimientos

Quiero expresar mi sincero agradecimiento

A mi Padre Celestial por permitirme estar en este tiempo y en este lugar, por las bendiciones y las pruebas, y por la oportunidad de progresar en este mundo.

A CONACyT por la confianza y el apoyo recibido durante mis estudios de maestría. Y al CINVESTAV por ser la institución que me ofreció los recursos para lograr esta meta.

A mi asesor, el Dr. Sergio V. Chapa Vergara y al M. en C. Noé Sierra Romero por sus consejos, y a cada uno de los docentes del departamento de computación por sus conocimientos transmitidos para formarme como profesional y como persona.

A mi esposa y compañera eterna, por estar en mi vida y por que con una mirada de tí, *Merith*, me siento el hombre más afortunado del universo.

A mis padres y hermanos, por estar conmigo desde el comienzo de esta carrera, por apoyarme y animarme a lograr mis metas.

A mis compañeros y a cada una de las personas que estuvieron conmigo durante este camino.

Índice general

Resumen	II
Abstract	III
Agradecimientos	IV
1. Introducción	1
1.1. Antecedentes	1
1.2. Objetivos	2
1.3. Organización de la tesis	3
2. Marco teórico	5
2.1. Sistemas de información geográfica	5
2.1.1. Escalas de mapas	12
2.1.2. Sistemas de coordenadas	13
2.2. El paradigma Web 2.0	13
2.3. Formatos gráficos Canvas y SVG	18
2.3.1. Canvas	18
2.3.2. SVG - <i>Scalable Vector Graphic</i>	19
2.4. Introducción a la programación basada en <i>tags</i>	21
3. Sistemas de bases de datos espaciales	24
3.1. Definición y clasificación de los datos espaciales	24
3.2. Estructura de los datos espaciales	25
3.2.1. El concepto de un dato espacial tipo <i>Geometry</i>	25
3.2.2. Definición de sistemas de bases de datos espaciales	29
3.3. Relación entre bases de datos espaciales y SIG	32
3.4. Introducción a <i>PostGIS</i>	35
4. Estado del arte	38
4.1. API para el desarrollo de sistemas de información geográfica	38
4.1.1. <i>Mapacel</i>	38
4.1.2. ArcGIS	40
4.1.3. GeoDjango	42
4.2. Sistemas de información geográfica en Web	42
4.2.1. Servidor de mapas	42

4.2.2.	Sistema de información turística con referencia particular a la ciudad de Huludao, China	43
4.2.3.	Sistema de información geográfica: <i>DataCrossingDSS</i>	44
4.2.4.	Programación automática de sistemas Web a partir de descripciones GML+x: caso de estudio SIG	46
5.	Diseño y desarrollo	50
5.1.	Metodología de diseño y desarrollo	50
5.2.	Análisis de componentes en un SIG	51
5.3.	Generación dinámica de objetos geográficos	52
5.4.	Modificación de requerimientos	54
5.4.1.	Arquitectura de la biblioteca	54
5.4.2.	Herramientas adicionales	56
5.5.	Diseño de sistemas con reutilización de componentes	56
5.5.1.	Sistema Web de información geográfica para el desarrollo sustentable en hidrología	57
5.5.2.	Sistema Web de información geográfica para la toma de decisiones en la infraestructura educativa	58
5.6.	Desarrollo e integración	59
5.6.1.	<i>Tags</i> JSP creados para su reutilización en SIG	61
5.7.	Modelado UML	68
6.	Validación de SIG Web utilizando la biblioteca GISWEBTAG	75
6.1.	Validaciones básicas	75
6.2.	Sistema Web de información geográfica para el desarrollo sustentable en hidrología	83
6.3.	Sistema Web de información geográfica para la toma de decisiones en la infraestructura educativa	87
7.	Conclusiones	91
7.1.	Conclusiones	91
7.1.1.	Impacto tecnológico en Web	91
7.1.2.	Impacto en tecnología móvil	92
7.2.	Contribución	92
7.2.1.	Contribución a los SIG	92
7.2.2.	Contribución a los OGIS (<i>Open Geographic Information Systems</i>)	92
7.2.3.	Contribución tecnológica al desarrollo sustentable e impacto social	93
7.3.	Trabajo a futuro	93
	Bibliografía	95

Índice de figuras

1.1. Áreas de investigación del proyecto	4
2.1. Taxonomía en sistemas de información	11
2.2. Arquitectura general de un sistema Web de información geográfica	11
2.3. Métodos de representar la escala de un mapa.	12
2.4. Evolución de los estándares Web	16
2.5. Modelos de aplicaciones Web síncronos y asíncronos	17
2.6. Código ejemplo de utilización de los <i>tags</i> JSP	22
3.1. El modelo de la OGC (<i>Geometry object model</i>)	26
3.2. La relación del concepto <i>Geometry</i> con otros elementos	27
3.3. Estructuras de datos topológica y cartográfica	29
3.4. Evolución de los conceptos de procesamiento de datos espaciales	31
3.5. Tipos de datos espaciales <i>Geometry</i> en <i>Oracle Spatial</i> y <i>DB2 Spatial Extender</i>	33
3.6. Tipos de datos espaciales <i>Geometry</i> en <i>ESRI Geodatabase</i> y <i>PostGIS</i>	34
4.1. Arquitectura general del framework <i>mapaccel</i>	39
4.2. Diferentes áreas de aplicación de ArcGIS	40
4.3. Diagrama de flujo del SIG para la ciudad de Huludao	44
4.4. <i>Framework</i> del SIG para la ciudad de Huludao	45
4.5. Distribución de los componentes en la Internet del sistema de <i>Data Crossing</i>	46
5.1. Etapas para el diseño y desarrollo orientado a la reutilización	51
5.2. Esquema general propuesto para la creación de gráficos en navegadores Web	54
5.3. Estructura de la biblioteca de <i>tags</i>	55
5.4. Diagrama de componentes del SIG en Web para el desarrollo sustentable en hidrología	58
5.5. Diagrama de componentes del SIG en Web para la toma de decisiones en la infraestructura educativa	59
5.6. Diagrama de caso de uso del desarrollador de aplicaciones Web GIS	69
5.7. Diagrama de actividad de la generación del objeto geográfico	70
5.8. Diagrama de secuencia de la generación del objeto geográfico	71
5.9. Diagrama de actividades de un SIG en Web	72
5.10. Diagrama de componentes de un SIG en Web genérico	73

6.1. Flujo de movimientos para acercar los objetos en la pantalla sensible al tacto del dispositivo móvil	78
6.2. Flujo de movimientos para alejar completamente la vista de los objetos geográficos	78
6.3. Imágenes de gráficos SVG en el navegador Web Safari del <i>iPodTouch</i> , acercamiento y comparación vs. la carta topográfica.	79
6.4. Gráficos en formato SVG, a) muestra la capa de las corrientes de agua (azul) vías de comunicación (negro) y referencia topográfica (cafe); b) además muestra las curvas de nivel (rojo).	80
6.5. a) Visualización de las curvas de nivel de una zona montañosa, generada con SVG y b) es la carta topográfica de la zona.	81
6.6. Visualización de todas las capas en SVG, mostrando la resolución en: a) formato SVG y b) imagen TIF.	81
6.7. Visualización de las capas de corrientes de agua (azul), curvas de nivel (cafe) y vías de transportación (negro).	82
6.8. Arquitectura del sistema Web de información geográfica en Web para el desarrollo sustentable en hidrología.	83
6.9. Mapa en donde se localiza un nodo de sensores	84
6.10. Información a detalle del nodo de sensores.	84
6.11. Proceso de búsqueda de localidades. a) Pantalla para iniciar búsqueda; b) y c) pantallas para mostrar como introducir nombre de localidad en el <i>iPodTouch</i> ; d) pantalla mostrando los nombres de las localidades sugeridas de la base de datos.	85
6.12. Esquema conceptual de las tablas para el caso de estudio del SIG Web para el desarrollo sustentable en hidrología.	86
6.13. Captura de pantalla del sistema en donde se visualizan ríos y una escuela.	87
6.14. En la Figura a) se muestra un acercamiento al mapa en formato Canvas e imagen cuando comienza a aparecer escuela y en b) su información.	88
6.15. Imagen de la escuela, datos toponímicos y del responsable.	88
6.16. a) Pantalla que muestra el lugar geográfico y la foto de la escuela, b) Atributos de la escuela u objeto geográfico.	89
6.17. Región con diversas escuelas y la capa de hidrología.	89

Índice de tablas

2.1. Ejemplos de sinónimos para el término Sistema de información geográfica y su fuente o motivo por el cual se deriva	7
2.2. Comparación del proceso cartográfico al aplicarlo a la cartografía tradicional y a sistemas de información geográfica	8
2.3. Analogía entre los subsistemas de entrada de dato de SIG digitalizados vs. mapas tradicionales	8
2.4. Analogía entre los subsistemas de almacenamiento y consulta de datos de SIG digitalizados vs. mapas tradicionales	9
2.5. Analogía entre los subsistemas de análisis de SIG digitalizados vs. mapas tradicionales	9
2.6. Analogía entre los subsistemas de visualización y reportes de SIG digitalizados vs. mapas tradicionales	10
3.1. La división de trabajo entre sistemas de bases de datos espaciales y SIG	32
5.1. Navegadores Web que soportan los formatos gráficos SVG y Canvas.	53

Capítulo 1

Introducción

1.1. Antecedentes

En la última década del siglo pasado surgió la *World Wide Web*, el protocolo HTTP y la creación de tecnologías Web. También a mediados de esa década se presenta el primer atlas en línea (*The National Atlas of Canada*, en 1994) y poco después comienza una búsqueda acelerada de servicios a través de la Internet. Por ejemplo, *Mapquest* [1] ofrece en 1996 los servicios de búsqueda de direcciones y rutas. Para el año 2003, NASA *World Wind* [2] es liberado, y dos años después *Google* presenta *Google Maps* [3], el cual ya permite añadir los servicios de mapas a sitios Web de los usuarios con la *Google Maps API*.

Con los avances en las tecnologías inalámbricas de transmisión de datos, la ubicuidad esta penetrando en nuestras vidas [4], y para sistemas de información geográfica, el reto es crear sistemas Web de información geográfica¹ que puedan ser accedidos desde dispositivos móviles [5].

Actualmente existen API para la visualización y desarrollo de SIG (Sistemas de Información Geográfica) como *mapaccel* [6], *ArcGIS* [7] y *GeoDjango* [8] que ofrecen un conjunto de herramientas para crear SIG, tanto en un ambiente Web, como en aplicaciones de escritorio, en el capítulo 4 se describen cada uno de ellos.

Por otra parte está el sistema *msCross* [9] como parte del proyecto de *DataCrossingDSS* que es una aplicación en Web 2.0/AJAX², pero no tiene una aplicación diseñada especialmente para dispositivos móviles.

El proyecto histórico inmediato se titula “Programación automática de sistemas Web a partir de descripciones GML+x: caso de estudio SIG” realizado por Hernández [10] que es una herramienta de software que permite la generación automática del sitio Web para la administración de un sistema de información geográfica con tecnologías como XML (*eXtensible Markup Language*), GML (*Geographic Markup Language*), XSLT (*eXtensible Stylesheet Language Transformation*) y SVG (*Scalable Vector Graphic*).

¹WebGIS, en inglés, *Web Geographic Information System*

²AJAX, por sus siglas en inglés, *Asynchronous JavaScript and XML*

En mi proyecto de tesis se hace uso de la tecnología del dispositivo móvil iPodTouch³, de los avances en SIG y de las tecnologías para Web 2.0 para plantear el diseño y desarrollo de una API para visualizar objetos geográficos y crear SIG en Web en dispositivos móviles. Ofreciendo al desarrollador módulos para la generación de objetos geográficos para ser visualizados en el navegador en formato Canvas y SVG. Se ofrece también un módulo para realizar consultas a la base de datos espacial. La API está formada por una biblioteca de *tags-JSP* y archivos en *JavaScript*.

Este trabajo pretende realizar un conjunto de herramientas que sirvan para crear Sistemas de Información Geográfica en la Web para dispositivos móviles y de escritorio. En el caso especial se desarrollo para los dispositivos móviles *iPodTouch/iPhone* de Apple. Y así crear sistemas que se caractericen por la rapidez en su operación y la conveniencia de mantenimiento y uso, un acceso desde dispositivos móviles en cualquier momento sin la necesidad de algún software SIG⁴ y haciendo uso de las tecnologías para Web 2.0. Además de un costo bajo en su desarrollo, uso y gestión.

1.2. Objetivos

Objetivo particular

Diseñar y desarrollar una API para la visualización de datos espaciales en sistemas Web de información geográfica para dispositivos móviles. Además, de crear dos sistemas para validar la biblioteca de *tags* JSP que se creó.

Objetivos particulares

- Crear una biblioteca de *tags* JSP para visualizar y consultar datos espaciales.
- La biblioteca de *tags* debe cumplir con las especificaciones del *Open GIS Consortium*, quien establece los estándares para sistemas de información geográfica.
- Generar automáticamente objetos geográficos en formato Canvas y SVG.
- Analizar los formatos gráficos Canvas y SVG para mostrar datos espaciales en el navegador Web para los dispositivos móviles *iPodTouch / iPhone*.
- La biblioteca de *tags* debe facilitar el desarrollo de sistemas de información geográfica en Web.

³Características generales: OS X, Wi-fi(802.11b/g), Pantalla amplia sensible al tacto de 480x320 pixeles y soporte para aplicaciones Web 2.0 con el navegador Web Safari

⁴Software que se instala en el dispositivo para que pueda funcionar el SIG

1.3. Organización de la tesis

El presente documento se divide en 7 partes en las que describimos el trabajo desarrollado para el diseño y desarrollo de una API para visualizar datos espaciales en sistemas de información geográfica en dispositivos móviles.

El capítulo 1 presenta una introducción al proyecto y al documento de tesis, además de plantear los antecedentes que dan auge a los sistemas Web de información geográfica en la actualidad. También se establecen los objetivos, el general y los particulares del proyecto.

El capítulo 2 presenta un marco teórico de las áreas de investigación que cubre el proyecto. Se definen conceptos y se realizan comparaciones en sistemas de información geográfica para clasificar estos sistemas. Y se explican las estructuras para formar objetos geográficos en formatos Canvas y SVG. En la Figura 1.1 se presentan de forma esquemática las áreas de investigación que se estudian en este proyecto.

El capítulo 3 explica con detalle que es un sistema de base de datos espacial para mostrar con énfasis las diferencias entre éste y un sistema de información geográfica y evitar confundir estos conceptos. También, le ofrece al lector los conceptos que servirán para comprender los próximos capítulos de este documento (*e.g.* datos espaciales, tipos de datos espaciales y la definición de los subsistemas que forman un sistema de información geográfica). Luego se presenta la comparación entre la cartografía tradicional y los sistemas de información geográfica para descubrir las ventajas que ofrecen estos últimos.

El capítulo 4 muestra los proyectos similares o proyectos en los cuales un módulo es similar a este proyecto. Primero se muestran API para desarrollo de sistemas de información geográfica. Después, se presentan sistemas de información geográfica y un proyecto para generar automáticamente sistemas de información. Cabe mencionar que estos proyectos desarrollan el subsistema de visualización en navegadores Web para PC, pero no lo hacen para dispositivos móviles especialmente.

El capítulo 5 explica el diseño y desarrollo del proyecto. Primeramente, se define la metodología de diseño y desarrollo, la cual es con un enfoque orientado a la reutilización de componentes. Después, se presentan los diseños de los sistemas de información geográfica para validar la API. También se modela la API y su uso a través de lenguaje UML con diagramas.

El capítulo 6 muestra los sistemas para validar la API, uno es para apoyar el desarrollo sustentable en hidrología y el otro es para dar soporte a la toma de decisiones en la infraestructura educativa. Además de mostrar validaciones y comparaciones básicas con las cartas topográficas en formato TIF.

El capítulo 7 presenta los resultados y conclusiones. Además de las contribuciones de este proyecto y a mi parecer, las líneas que este proyecto podría seguir en un futuro.

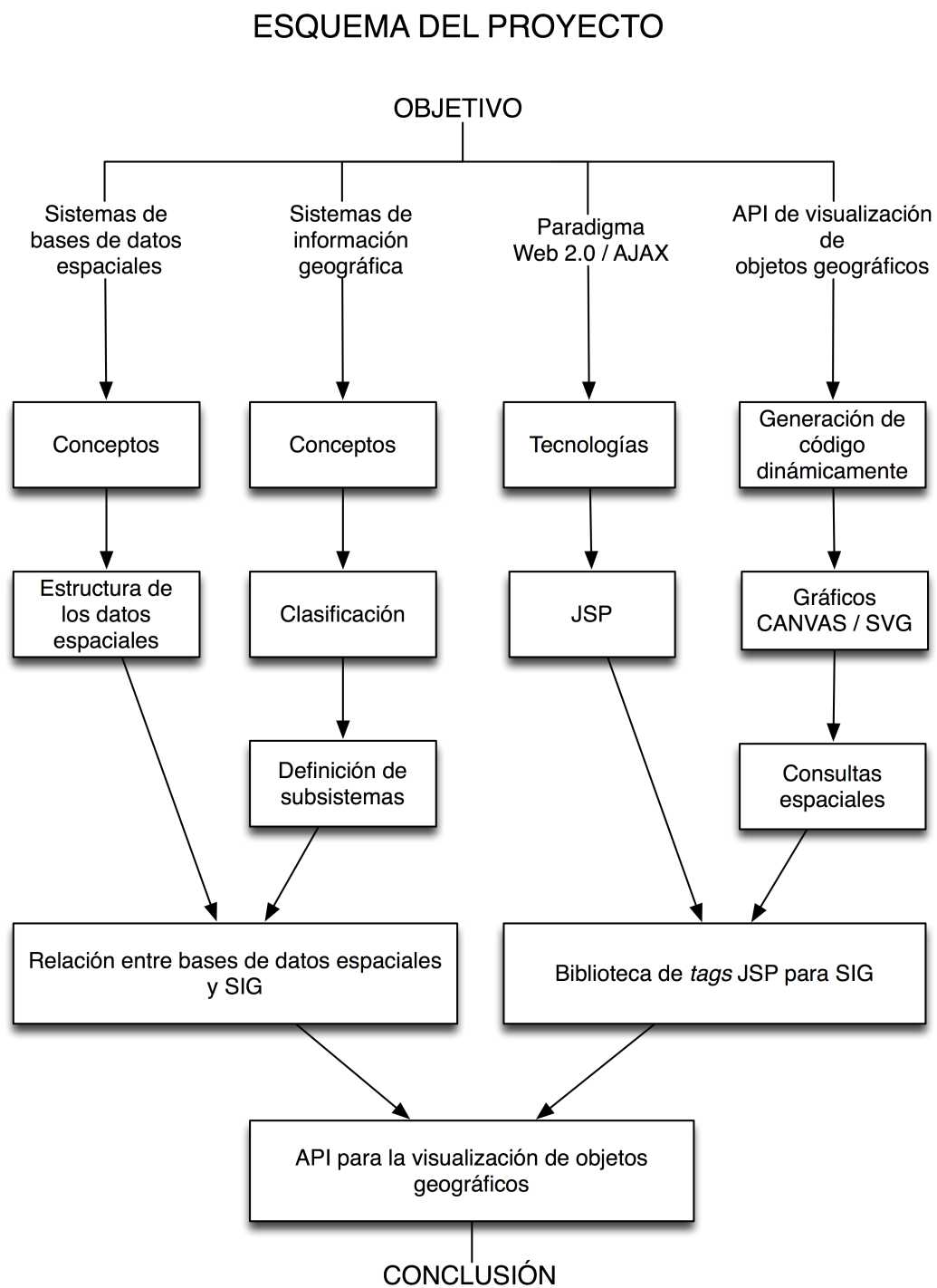


Figura 1.1: Áreas de investigación del proyecto

Capítulo 2

Marco teórico

2.1. Sistemas de información geográfica

A principios de la década de 1960, el departamento de desarrollo rural y silvicultura de Canadá decidió desarrollar un proyecto para gestionar los recursos de su territorio. Como resultado de las primeras tareas se obtuvieron el inventario y mapeo de los bosques, los recursos minerales, los requerimientos de la fauna en estado salvaje y la calidad y disponibilidad del agua [11]. Tan solo el objetivo de obtener los límites geográficos es sorprendente, se requiere de grandes cantidades de datos. Además de su compilación, evaluación, análisis y modelado.

El trabajo de obtener los datos es realizado de forma manual, lo que implica que se debe cubrir toda zona que se debe mapear, lo cual no se puede hacer de un vistazo. Así que, cuando es una área demasiado grande se necesita de muchos cartógrafos y grandes cantidades de tiempo y dinero.

En una reunión entre el director del departamento de desarrollo rural y silvicultura¹ de Canadá y un ejecutivo de IBM resultó la idea de crear una aplicación para resolver el problema con las tecnologías emergentes de IBM. Entonces se funda la División para la planeación de sistemas de información para la planeación regional por el gobierno Canadiense. Allí es cuando se crea el primer SIG conocido, *Canada Geographic Information System* (CGIS) y su primer tarea fue clasificar y mapear los recursos de tierra de Canadá.

Hoy en día, se ha incrementado la necesidad de realizar mapeo a grandes escalas y operaciones de análisis sobre mapas digitales, a continuación se enlistan algunas aplicaciones:

- los silvicultores desean mantener actualizado el inventario de los bosques, y un SIG ofrece herramientas para la administración, actualización y operaciones sobre de los datos,
- el departamento de bomberos tiene la necesidad de un SIG para mejorar las rutas de traslado al lugar del siniestro, y por consiguiente, asegurar una respuesta de atención

¹Ciencia que trata de cultivo de bosques o montes según la Real Academia Española.

rápida,

- la milicia puede utilizar SIG para determinar planes de batalla apropiados y organizar movimientos de tropas
- las compañías telefónicas celulares que desean proveer un mejor servicio, deben evitar barreras entre la visualización de línea de vista entre sus estaciones base y los usuarios para una adecuada transmisión, estas barreras pueden ser construcciones cuya geometría pueden administrarse y analizarse mediante un SIG,
- el gobierno puede desarrollar un SIG para planear el crecimiento demográfico o modificar las regulaciones para algunas zonas para evitar altos índices de contaminación,
- en negocios los SIG pueden usarse para enviar publicidad de productos a ciertos consumidores potenciales, tomando como criterio una consulta espacial sobre la ubicación de los consumidores,
- el departamento de policía usa SIG para compilar información y caracterizar los movimientos y operaciones de sospechosos o bandas delictivas,
- las disciplinas como la geografía, biología, geología, arquitectura, entre otras tiene las tecnologías ahora para desarrollar y probar hipótesis sobre patrones de fenómenos naturales en el globo terráqueo.

El potencial de usuarios de SIG es cada vez mayor, en donde el tipo y número de usuarios está creciendo en pasos logarítmicos. Este crecimiento es indicativo de que los SIG son una tecnología poderosa y en constante desarrollo.

Los SIG modernos han mejorado la utilidad de los mapas, pueden ser analizados automáticamente. Los SIG han cambiado la forma en la cual se ocupan los mapas tradicionales, ahora al enfocarnos en el contexto de SIG, vemos los mapas como un conjunto de datos espaciales que son recolectados y compilados. Tareas que eran imposibles con los mapas tradicionales ahora se pueden realizar. Las organizaciones, sean públicas o privadas, han llegado a incorporar SIG, por lo que la demanda se incrementará por parte de los usuarios quienes ya tendrán los conceptos de SIG detrás de la tecnología que utilicen.

Entonces se define un SIG como:

una herramienta que permite el procesamiento espacial de datos en información, que generalmente es explícita y se utiliza para la toma de decisiones sobre áreas geográficas. Esta definición puede no ser comprensiva y es debido a que se trabaja demasiado en las áreas que comprende y depende demasiado de la integración de estas áreas.

El término SIG por sí mismo llega a ser hibridado o modificado de acuerdo el intelecto, la cultura, economía y aún de objetivos políticos de quien lo refiera (véase Tabla 2.1).

Un SIG, contiene los siguientes subsistemas:

Terminología	Fuente
<i>Geographic information system</i>	Estados unidos
<i>Geographical information system</i>	Europa
<i>Geomatique</i>	Canada
<i>Georelational information system</i>	terminología basada en tecnología
<i>Natural resources information system</i>	terminología basada en disciplina
<i>Geoscience or geological information system</i>	terminología basada en disciplina
<i>Spatial information system</i>	derivativo de no-geográfico
<i>Spatial data analysis system</i>	terminología basada en lo que hacen los sistemas

Tabla 2.1: Ejemplos de sinónimos para el término Sistema de información geográfica y su fuente o motivo por el cual se deriva

1. una entrada de datos que reúne y preprocesa datos espaciales de varias fuentes. Este subsistema es responsable de transformar de diferentes tipos de datos a datos espaciales (i.e., de símbolos de elevación en un mapa cartográfico a puntos de elevación dentro de un SIG),
2. un subsistema de almacenamiento y consultas de datos que gestione los datos espaciales de manera que permita editarlos, actualizarlos y consultarlos,
3. un subsistema de manipulación y análisis de datos que realice operaciones sobre ellos; agregue o elimine; estime parámetros y limitaciones; y aplique funciones,
4. un subsistema de visualización que despliegue todos los datos o parte de ellos de forma tabular, gráfica o en un mapa.

La definición de subsistemas permite la comparación entre SIG modernos digitalizados completamente y su contraparte, los SIG con mapas tradicionales, particularmente cuando se considera el proceso cartográfico (véase tabla 2.2). El primer subsistema SIG, la entrada de datos espacial, es equivalente al primer y segundo paso del proceso cartográfico que realiza las tareas de reunir los datos y compilar los mapas [12] (véase tabla 2.3). En la cartografía tradicional los cartógrafos compilan y almacenan puntos, líneas y áreas en un medio físico como papel. Los datos son recolectados de fuentes tales como fotografías digitales, sensado digital remoto, entrevistas, descripciones visuales, censos y datos estadísticos.

Por otra parte, en SIG digitalizados se utilizan dispositivos electrónicos para almacenar o codificar puntos, líneas, y áreas en computadoras. Las fuentes de datos son frecuentemente las mismas para el proceso de mapeo. Ahora también se incluyen fuentes de datos digitales como: gráficos de líneas digitales (*digital line graphs, DLG*), modelos de elevación digital (*digital elevation models, DEM*) y fotos "ortoquads" (*digital orthophotoquads*), entre otras.

MAPA TRADICIONAL	SIG
Colección de datos: fotos aereas, encuestas, etc.	Colección de datos: fotos aereas, entrevistas, etc.
Procesamiento de datos: agregación, clasificación, etc.; proceso lineal	Procesamiento de datos: agregación, clasificación, análisis detallado; proceso circular
Producción de mapas: paso final excepto para reproducir y diseminación	Producción de mapas: no siempre es el paso final; comúnmente un mapa es usado para producir otros mas.
Reproducción de mapas	Reproducción de mapas

Tabla 2.2: Comparación del proceso cartográfico al aplicarlo a la cartografía tradicional y a sistemas de información geográfica

MAPA TRADICIONAL	SIG
Almacenar la entrada (compilar) en papel desde una fuente de datos espaciales: puntos, líneas y áreas	Entrada: Codificada en la computadora desde un conjunto de fuentes de datos espaciales: puntos, líneas y áreas
Fuentes: fotografías aereas, sensado remoto digital, encuestas, descripciones visuales, datos de censos y datos estadísticos.	Fuentes: las mismas que los mapas tradicionales y gráficos de líneas digitales (DLG), Modelos de elevación digital (DEM), fotos "ortoquads" digitales y otras bases de datos digitales.

Tabla 2.3: Analogía entre los subsistemas de entrada de dato de SIG digitalizados vs. mapas tradicionales

Para el caso del segundo subsistema SIG, referente al almacenamiento y consultas, no hay un proceso contrario en el método cartográfico tradicional, el mapa por sí mismo almacena datos y permite consultarlos. Las primitivas de los objetos espaciales, puntos, líneas y áreas tiene que ser colocadas en un documento cartográfico y pueden ser leídas por las personas. Se dice que el mapa es la forma más compacta de almacenar información espacialmente relacionada y la forma más compleja podría ser cuando se preparan los datos espaciales para visualizarse gráficamente en dispositivos electrónicos.

El subsistema de almacenamiento y consulta de un SIG digitalizado tiene la ventaja, contra los mapas gráficos en que las consultas son específicamente la información indi-

MAPA TRADICIONAL	SIG
Puntos, líneas y áreas son dibujadas en papel con símbolos	Puntos, líneas y áreas son almacenadas como un par de coordenadas y puntos en la computadora
La consulta de datos se realiza simplemente la lectura del mapa	Los atributos de las tablas son asociados con el par de coordenadas la consulta requiere técnicas eficientes de búsqueda

Tabla 2.4: Analogía entre los subsistemas de almacenamiento y consulta de datos de SIG digitalizados vs. mapas tradicionales

cada (véase tabla 2.4). Y con este formato se enfatiza la formulación de consultas y la obtención de respuestas apropiadas. En términos generales, este subsistema almacena los objetos espaciales a través de puntos, líneas y polígonos. Dichos objetos son asociados a su toponimia² y características.

En el subsistema SIG de análisis, una vez más no hay un método cartográfico que se pueda comparar, excepto que el mapa es una herramienta fundamental para el análisis de datos relacionados espacialmente (véase Tabla 2.5). Para los mapas cartográficos se utilizan reglas para medir distancias, ángulos para encontrar direcciones y planímetros para medir áreas [13]. Los cartógrafos tienen necesidad de analizar mapas, para comparar y contrastar patrones de fenómenos relacionados con el globo terráqueo. El análisis en SIG aprovecha la potencia de procesamiento de las computadoras modernas para medir, comparar y describir el contenido de las bases de datos. Esto permite tener acceso a datos y permite la agregación para diversos análisis.

MAPA TRADICIONAL	SIG
Requiere: reglas, planímetros compases y otras herramientas, todas utilizadas por un analista	Usa el potencial de las computadoras para medir, comparar y describir contenido de la base de datos
Se limita en que los datos son agregados representados en el mapa de papel	Permite tener acceso a los datos y permite reclasificar para futuros análisis

Tabla 2.5: Analogía entre los subsistemas de análisis de SIG digitalizados vs. mapas tradicionales

La principal diferencia entre SIG digitalizados y mapas cartográficos, después de ana-

²Nombres propios de los elementos que aparecen en un mapa

lizar las analogías mostradas en las tablas, es el énfasis que hacen los SIG en la forma de reportar el resultado del subsistema de análisis de datos (véase tabla 2.6).

MAPA TRADICIONAL	SIG
Sólo un gráfico	El mapa es tipo de salida de un SIG
Varias formas de mapas	Con pocas excepciones, un SIG las misma opciones que los mapas dibujados a mano
Las modificaciones pueden incluir cartogramas	También incluyen tablas diagramas, fotografías, entre otros.

Tabla 2.6: Analogía entre los subsistemas de visualización y reportes de SIG digitalizados vs. mapas tradicionales

En la Figura 2.1 se muestra la taxonomía que presenta la separación clara entre sistemas de información espaciales y no espaciales. El concepto de sistema de información geográfica aplica correctamente a sistemas de información espacial. Dentro de esta categoría hay dos clases: geográficos y no geográficos. Los sistemas de información espaciales no geográficos, están frecuentemente tratando con pequeñas porciones de espacio geográfico y no tienen una localización de referencia con el globo terráqueo. Así entonces, los sistemas CAD (*Computer Aided Drafting*), que sirven para visualizar y editar datos espaciales son débiles en el análisis, y se tipifican como sistemas de información no geográficos. Los sistemas de información geográfica tienen también una división:

1. los sistemas de información terrestres (*Land information system (LIS)*) y
2. los sistemas de información no terrestres (*Land information system (LIS)*) u otros SIG.

Aunque parezca un poco artificial, la separación se enfoca principalmente en las áreas terrestres que han sido geocodificadas y que han sido afectadas o que podrían serlo por factores relacionados a áreas terrestres. Estas utilizan censos de población, las actividades económicas y de casa, más que a la repartición y uso de la tierra. Por otra parte están los LIS, que se basan en la gestión de áreas terrestres.

En los últimos 40 años, los SIG han sido desarrollados utilizando una gran variedad de arquitecturas de computadoras [14]. Los primeros SIG se implementaban en mini-computadoras, luego en estaciones de trabajo, y después en PC, y ahora, los diseñadores recomiendan sistemas basados en Web [15] con acceso desde cualquier dispositivo sea fijo o móvil.

En la Figura 2.2 se muestra la arquitectura general de un sistema Web de información geográfica. Una capa en un SIG es el conjunto de datos u objetos espaciales que se caracterizan de acuerdo a un tópico específico (*e.g.* carreteras, zonas urbanas, parques, etc.). En un SIG existen diversas capas sobrepuestas, y todas georeferenciadas al mismo sistema de coordenadas y así poder realizar consultas y obtener información entre diferentes capas.

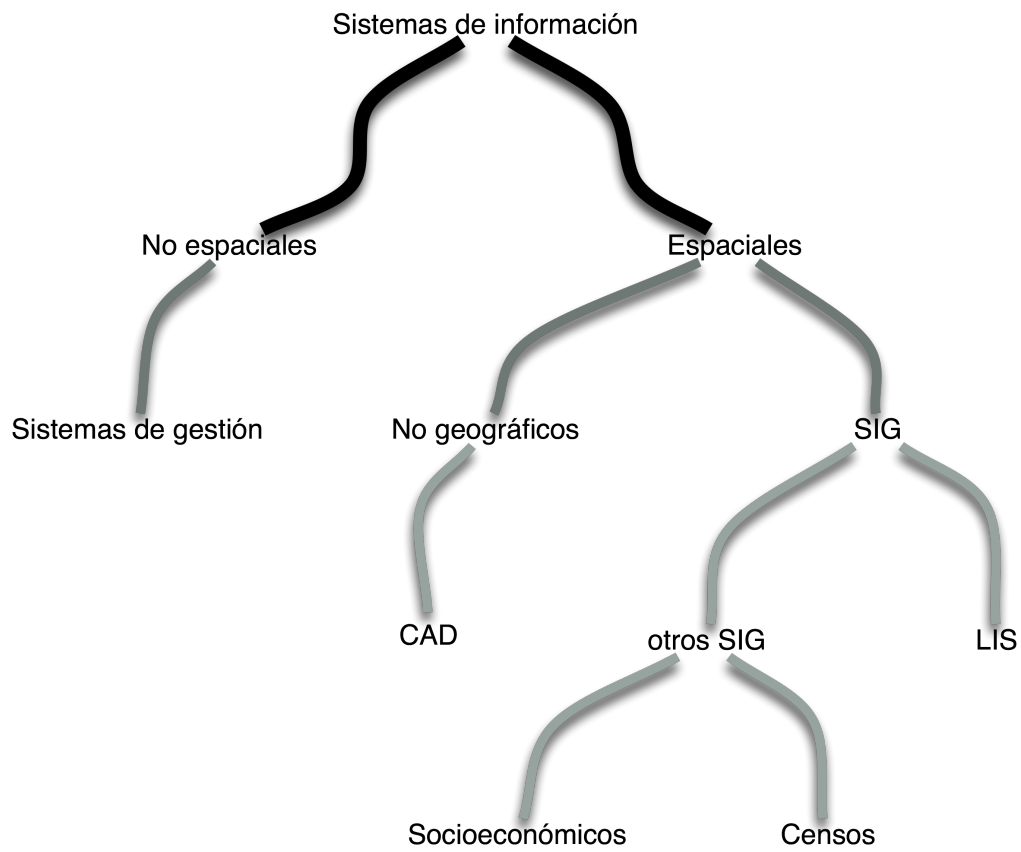


Figura 2.1: Taxonomía en sistemas de información

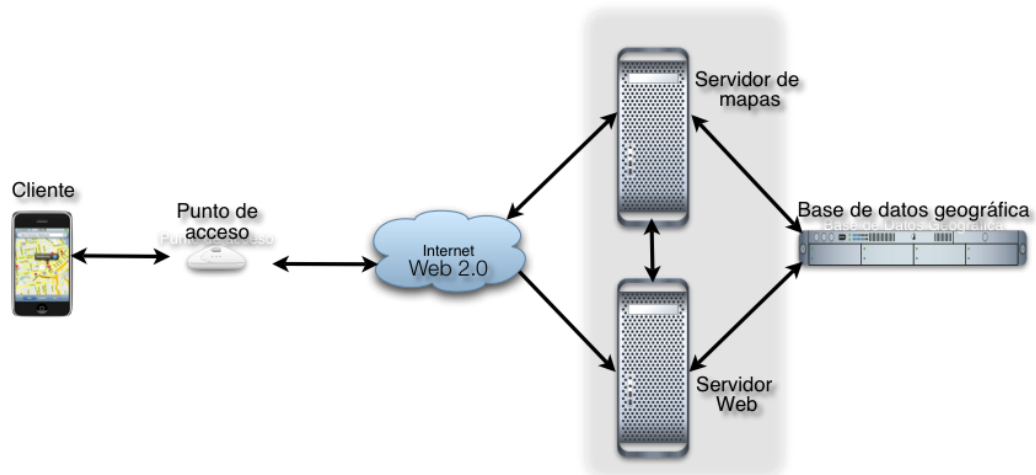


Figura 2.2: Arquitectura general de un sistema Web de información geográfica

Las división de los objetos espaciales en capas ayuda a organizar una sistema de base de datos espacial.

2.1.1. Escalas de mapas

Se debe recordar que los mapas son reducciones de la realidad, el propósito principal de cualquier mapa es mostrar a detalle, una región grande a simple vista, sin distracciones. La cantidad de reducción está en función del nivel de detalle que necesitemos para examinar el área. Si queremos observar una área muy pequeña, como la de un campo, no se requiere reducir mucho de la realidad como si tuvieramos que estudiar una área de 1000 km².

La escala, es el término aplicado a la cantidad de reducción encontrada en mapas (véase Figura 2.3). Es fácil definirla como la razón entre la distancia en el mapa y la distancia que es reamente en el globo terráqueo. Por ejemplo, una etiqueta en el mapa podría indicar que 1 pulgada en el mapa es equivalente a 63 360 pulgadas en la tierra. Una escala expresada como “1 pulgada es equivalente a 63 360 pulgadas” es llamada una escala verbal. Este método para expresar la escala tiene la ventaja de ser fácilmente comprendida por las personas que usen el mapa. Otro método común es el llamado fracción representativa (*representative fraction*, RF) en el cual ambas distancias, la del mapa y la del globo terráqueo, están dadas en la misma unidad, como una fracción. Cuando una dependencia ofrece mapas o fotografías aereas en una escala 1:600, se esperaría que sea un mapa muy detallado. Sin embargo, lo que indica la escala en el mapa es que 1 pulgada es equivalente a 600 pies. Esto se traduce al método de fracción representativa a 1:7200.



Escala gráfica o de barra

1" = 36 millas

Escala verbal

1:36,500

Fracción representativa

Figura 2.3: Métodos de representar la escala de un mapa.

La escala gráfica es otro método de expresar escalas, en éste las distancias reales aparecen explícitamente en el mapa. Tiene la ventaja de ser fácilmente transferible de una escala a otra. Pero con la desventaja de es difícil comunicar la escala al usuario de forma sencilla.

Es bueno mantener en mente que en una escala muy pequeña, digamos 1:100 000, una línea de un milimetro equivale a 100 000 milímetros o 100 metros en el globo terráqueo.

2.1.2. Sistemas de coordenadas

Una propiedad básica de los SIG, a diferencia de otros sistemas de información, es que los datos almacenados están georeferenciados, esto significa que los datos tienen definida su localización por un sistema de coordenadas referenciado.

Hay dos aspectos principales en los sistemas de coordenadas:

- El primero es la **consistencia**, ya que podría haber diferencias entre los sistemas de coordenadas usados por dos conjuntos de datos y reunirlos ambos en el mismo sistema, lo cual es un paso necesario antes de comenzar un análisis entre los datos.
- El segundo es el **computacional**, debido a que cada método para expresar coordenadas tiene sus limitaciones en lo difícil con que sus operaciones básicas (tales como distancia entre puntos o áreas de polígonos) son obtenidas y qué tanto los valores obtenidos podrían ser distorsionados a los del mundo real.

Entre los sistemas de coordenadas más comunes están: el sistema de coordenadas geográficas: latitud-longitud, es el más común y determina la localización de un punto en el globo terráqueo usando la latitud y la longitud. Se basa en una malla creada por meridianos y paralelos, los meridianos son las líneas que conectan de norte a sur los polos y los paralelos son las líneas que forman círculos alrededor de la tierra en paralelo con el ecuador. La longitud de un punto es definida como un ángulo entre su meridiano y el primo meridiano (0° este, pasando por el Observatorio de Greenwich, Londres, Inglaterra). La latitud de un punto es definida como un ángulo entre la normal a el esferoide pasando por un punto dado en el plano del ecuador. La longitud está entre $0-180^\circ$ al este desde su primo meridiano y $0-180^\circ$ al oeste. La latitud está entre $0-90^\circ$ al norte y $0-90^\circ$ al sur del ecuador (e.g. Murcia, España : -1.167° , 38.0° , en decimal y $1:10:00W$, $38:00:00N$ en sexagesimal) ; el sistema UTM (*Universal Transversal Mercator*) está basado la proyección (cilíndrica) con líneas (zonas) de norte a sur y de este a oeste como estándar en la proyección Mercator. UTM divide la tierra en 60 zonas con un ancho de 6° de longitud, numerados del 1 al 60. Cada zona está así mismo dividida con líneas a 8° de latitud con letras asignadas desde la C hasta la X, omitiendo las letras I y O.

2.2. El paradigma Web 2.0

La frase “Web 2.0” fue utilizada por primera vez el 2004 en O’Reilly Media [16], refiriéndose a la Internet como plataforma, y de igual forma al compartir servicios y crecer colaborativamente, usando redes sociales, como *wikis*.

Otra palabra clave en el paradigma Web 2.0 es *mashups*, cuya definición más simple es la combinación dinámica de datos a partir de diversas fuentes y múltiples formatos que están en la Web, por ejemplo, combinar en una aplicación Web basada sobre API de mapas (*Google Maps*), de fotos (*Flickr*), de comercio electrónico (*eBay*), entre otras, con el fin de compartir servicios y crear aplicaciones Web dentro del paradigma de la Web 2.0.

Una de las ventajas que ofrece crear aplicaciones Web sobre el paradigma Web 2.0, es que los usuarios sientan que están haciendo uso de una aplicación de escritorio y no de una típica aplicación Web. Lo anterior se logra mediante el uso de las tecnologías Web utilizadas en Web 2.0, las cuales permiten de una forma asíncrona realizar consultas, enviar peticiones y respuestas con el protocolo HTTP entre el servidor y el cliente, evitando así los momentos de espera para ejecutar tareas en el navegador Web.

Además se proponen principios para desarrollar aplicaciones Web 2.0 [16], en la siguiente lista aparecen los principios que se enfocan a este trabajo:

- **La Web como plataforma-** Las aplicaciones deberían tomar las ventajas de la Web como plataforma y hacer presencia en la Web. Trabajando de forma correlacionada con la gran apertura y conectividad que existe a la Web, ofreciendo servicios alcanzables a todos los usuarios.
- **Modelos de programación ligera-** Hay preferencia a utilizar protocolos simples, como REST (*Representation State Transfer*), SOAP (*Simple Object Access Protocol*) y RSS (*Really Simple Syndication*).
- **Desarrollo para diversos dispositivos-** En un mundo en el cual se puede acceder a la Internet desde cualquier dispositivo, hoy en día el que un dispositivo tenga un navegador Web ya no es opción. Las aplicaciones Web interactúan con una gran variedad de dispositivos que acceden a Internet, entre estos están, desde computadoras de escritorio, portátiles hasta dispositivos más especializados como los PDA y iPod.
- **Experiencia enriquecida al usuario-** Los servicios tienden a ser RIA (*Rich Internet Application*), lo cual da dinamismo e interacción con los usuarios. AJAX es una solución para ofrecer este tipo de aplicaciones.

Tecnologías AJAX

AJAX (*Asynchronous JavaScript and XML*) es un término acuñado por J. James Garrett en febrero de 2005 en su artículo llamado "AJAX: a new approach to Web applications"[17]. AJAX es un conjunto de técnicas de programación, estas técnicas se aplican a una página o sección de esa página Web para que se actualice de forma inmediata al detectar eventos generados por el usuario o por la misma página Web. Garret en su artículo enfatiza un mundo en donde las aplicaciones Web pueden ser como las aplicaciones creadas para PC en su funcionalidad, simplicidad, respuesta en tiempo, y en su contenido de interfaces amigables a los usuarios finales.

El paradigma Web 2.0 con AJAX incorpora los siguientes lenguajes y tecnologías [18][19]:

XML (*eXtensible Markup Language*) - XML es el formato de los datos que regresan del servidor en respuesta a una petición asíncrona del navegador Web. La respuesta XML es procesada por código *Javascript* en el navegador, causando cambios en el documento HTML, mediante la manipulación del DOM o del código CSS,

XMLHttpRequest - Este objeto fue creado para permitir a los desarrolladores iniciar peticiones HTTP desde cualquier aplicación. Esta petición fue diseñada para regresar un documento XML. Así el objeto *XMLHttpRequest* provee una forma fácil de acceder a la información a través de un documento XML. Una vez creado el objeto *XMLHttpRequest*, se pueden comenzar peticiones HTTP a partir de código *JavaScript*, los parámetros de entrada para comenzar son: el tipo de petición (*GET* o *POST*), la URL para enviar la petición y una bandera para indicar si la petición debe ser asíncrona,

JavaScript - Es un lenguaje de programación que puede ser embebido dentro de documentos HTML. Las funciones pueden ser ejecutadas automáticamente cuando se abre una página Web, en respuesta a eventos HTML o disparados por eventos del navegador Web,

HTML/XHTML (*Hypertext Markup Language*) - Para la presentación basada en estándares en los navegadores Web,

DOM (*Document Object Model*) - DOM es la representación de la estructura de árbol de los objetos de un documento HTML en el navegador Web. Manipulando el DOM con *Javascript* se logra una interacción dinámica con el documento HTML que puede ser modificado y visualizado, sin la necesidad de recargar la página Web,

CSS (*Cascading Style Sheets*) - Es un lenguaje con el objetivo de simplificar el diseño y estilo de páginas Web. CSS permite controlar el color del texto, el estilo de las letras el espaciado entre párrafos, el diseño de la tablas y una variedad de efectos visuales. Comúnmente, CSS es combinado con los lenguajes XML y XHTML. Al mantener la hoja de estilo (CSS) separada de la estructura HTML, el código puede ser modificado consistentemente y fácilmente. Similarmente a DOM, CSS puede ser modificado vía código *Javascript* dinámicamente cambiando el formato y estilo sin recargar la página Web.

Para enfatizar la evolución de los estándares Web, desde el comienzo del lenguaje HTML hasta hoy en día se muestra en la Figura 2.4 un panorama general. En la misma Figura las líneas continuas entre los rectángulos representan la existencia de una relación entre lenguajes o tecnologías y la línea discontinua entre el lenguaje XML y AJAX define la estrecha relación y dependencia de Ajax con XML. Se muestra primeramente HTML como una base de otros estándares que emergen de él. Además, está el lenguaje XML, que da soporte a diversos proyectos en los cuales se utiliza éste como base, creando el "Mundo XML". Ambos lenguajes se reúnen para forman XHTML. Para añadir hojas de estilo y elementos con lenguajes de programación como *JavaScript* aparece DHTML a partir de HTML con la añadidura de dinamismo ("*D*" *Dynamic*) y finalmente se llega a las tecnologías Web 2.0 con AJAX.

Envío y recepción asíncrona de datos utilizando las tecnologías XML

Los usuarios interactúan con las aplicaciones Web clásicas como un flujo de trabajo. En esa interacción hay peticiones constantes de información hacia el servidor Web, que responde de forma síncrona con el envío de nuevas páginas al navegador Web del cliente,

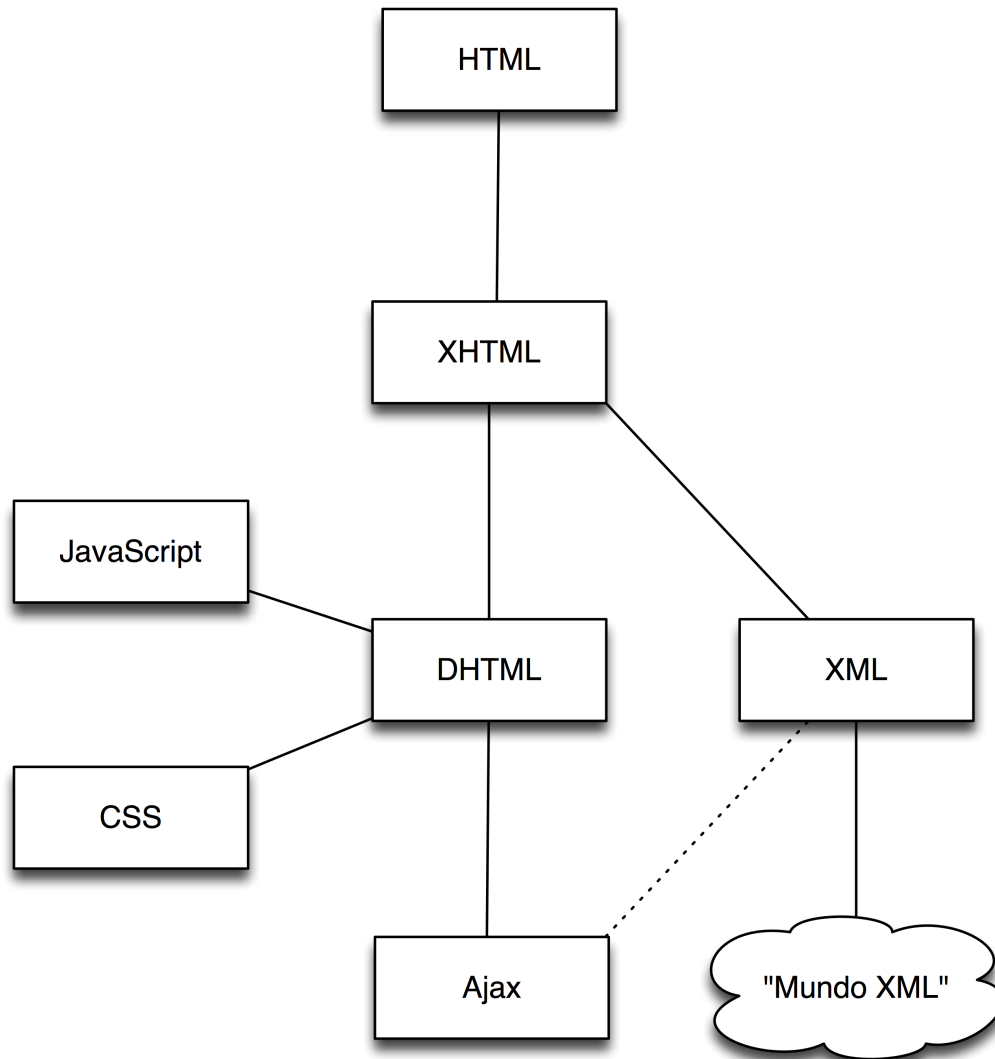


Figura 2.4: Evolución de los estándares Web

lo que deriva en un tiempo de espera e interrupciones al usuario, en donde se considera que la mayoría de las peticiones al servidor son lentas debido a la latencia de las redes. Entonces, es cuando las ventajas de la comunicación asincrónica entre cliente y servidor mejoran la interacción entre el usuario y la aplicación Web. Debido a una interfaz que responde a eventos generados por el usuario en el navegador Web, o del mismo sistema Web para ejecutar esas tareas de forma transparente al usuario. Recientemente, el objeto *XMLHttpRequest* ofrece una forma más clara y una solución para manejar comunicación asincrónica entre cliente y servidor.

Los objetos *XmlDocument* y *XMLHttpRequest* son unas extensiones no estandarizadas soportadas por los navegadores Web. Estos objetos manejan el proceso de realizar llamadas asíncronas considerablemente, debido a que fueron diseñados explícitamente para recuperar datos de forma asincrónica. Ambos objetos originaron la liberación del compo-

nente *Microsoft-specific ActiveX* que está disponible como objetos de *JavaScript* en el navegador Internet Explorer.

Otros navegadores Web han implementado una funcionalidad similar con objetos nativos. Los objetos *XMLHttpRequest* proporcionan un control más fino sobre las peticiones.

El modelo de aplicaciones AJAX

En un comienzo, la Web solamente tenía la intención de formatear y presentar documentos HTML. Lo que se conocen como aplicaciones Web “clásicas”, el usuario introduce los datos, envía la página al servidor y espera por la respuesta [20]. Al usar la Web como plataforma, la sincronía significa que el usuario realiza peticiones de páginas HTML, y el navegador web envía un petición HTTP (*HTTP request*) a un servidor Web. El servidor procesa las peticiones, entonces regresa una respuesta a el navegador Web en forma de una página HTML. El navegador muestra la página HTML requerida. El servidor Web nunca inicia peticiones, la comunicación siempre es de un sólo camino. El ciclo “petición / respuesta” es síncrono, durante el lapso que el usuario está esperando (véase Figura 2.5).

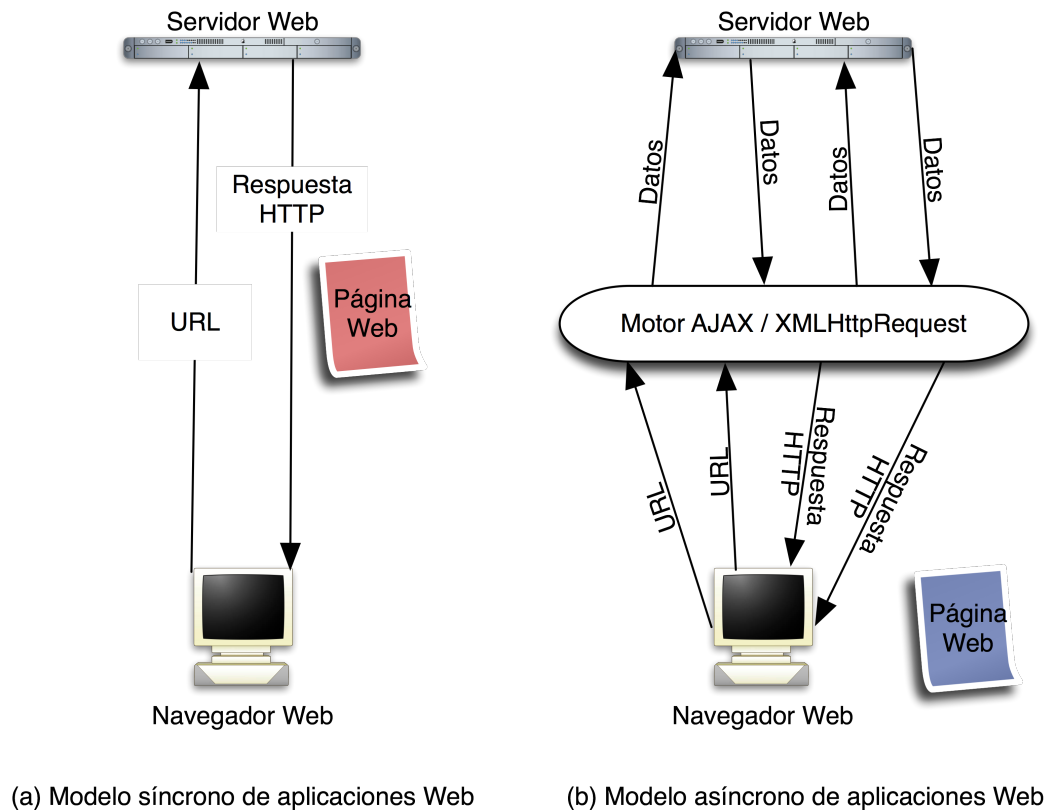


Figura 2.5: Modelos de aplicaciones Web síncronos y asíncronos

2.3. Formatos gráficos Canvas y SVG

En esta sección se describen las características principales de los formatos gráficos Canvas y SVG (*Scalable Vector Graphic*). Además de pequeños ejemplos de código para formar gráficos.

2.3.1. Canvas

El *tag* Canvas (“<canvas>”) es un elemento de HTML el cual puede ser utilizado para dibujar gráficos usando *scripts* (usualmente JavaScript). Se utiliza para dibujar gráficos, crear composiciones de fotos o animaciones. El *tag* Canvas fue ocupado primeramente por Mac OS X en su aplicación llamada *Dashboard* y después en el navegador Web Safari. El *tag* Canvas es parte de la especificación *WHATWG³ Web applications 1.0*, también conocida HTML 5 [21] .

Usar el *tag* “<canvas>” no es muy complicado, pero es necesario saber lo básico de HTML y *JavaScript*. Algo a considerar es que el elemento asociado al *tag* “<canvas>” no es soportado en versiones antiguas de navegadores Web, así que se necesita actualizar a las últimas versiones para visualizar gráficos en Canvas.

Ejemplo de código fuente para generar un gráfico con el elemento <canvas>:

```
(1)<html>
(2) <head>
(3)  <script type="application/x-javascript">
(4)    function draw() {
(5)      var canvas = document.getElementById("canvas");
(6)      if (canvas.getContext) {
(7)        var ctx = canvas.getContext("2d");
(8)        ctx.fillStyle = "rgb(200,0,0)";
(9)        ctx.fillRect (10, 10, 55, 50);
(10)       ctx.fillStyle = "rgba(0, 0, 200, 0.5)";
(11)       ctx.fillRect (30, 30, 55, 50);
(12)     }
(13)  }
(14) </script>
(15) </head>
(16) <body onload="draw();">
(17)  <canvas id="canvas" width="150" height="150"></canvas>
(18) </body>
(19)</html>
```

Descripción del código

En las líneas (2)-(15) se declara la etiqueta <head> que contiene los elementos que se

³Web Hypertext Application Technology Working Group, portal Web: <http://www.whatwg.org/>

cargan cuando se abre la página Web, en este caso, en la línea (3) se define un *script* que contiene lenguaje JavaScript y a partir de la línea (4) se define la función que forma el gráfico Canvas, en la línea (5) se declara una variable llamada “*canvas*” que almacena el elemento con identificador “*canvas*” (línea (17)) del cuerpo del documento (líneas (16)-(18)), con la variable se define un gráfico en dos dimensiones (línea (7)), en la línea (8) se declara el color para un rectángulo que se sitúa en el plano con un valor de 10, tanto en el eje *x*, como en el eje *y*; además tiene una altura de 55 unidades y un ancho de 50. En las líneas (10) y (11) también se crea un rectángulo; pero a diferencia del anterior, que en éste se define un nivel de opacidad de 0.5 en la línea (10). Al cargar el cuerpo del documento se acciona el evento *onload* definido en la línea (16) para que ejecute la función *draw()* para desplegar el gráfico Canvas dentro del *tag* Canvas establecido en la línea (17). En el *tag* Canvas también se define el ancho y alto del espacio para dibujar.

2.3.2. SVG - Scalable Vector Graphic

SVG es un lenguaje para describir gráficos en dos dimensiones a través de una estructura XML. SVG permite tres tipos de objetos gráficos: formas vectoriales gráficas (*e.g.* rutas que consistan de líneas y curvas), imágenes y texto. Los objetos gráficos pueden ser agrupados, transformados, estilizados y combinados con otros objetos. El conjunto de características incluye transformaciones, efectos de con filtros, máscaras y plantillas de objetos entre otros.

Los gráficos SVG pueden ser interactivos o dinámicos. Las animaciones pueden ser definidas o generadas declarativamente (*i.e.*, incrustando los elementos de la animación SVG en el cuerpo de un SVG) o vía un *script* que la genere.

En las aplicaciones más sofisticadas con SVG es posible utilizar *JavaScript* como lenguaje para acceder al DOM SVG (*i.e.*, al modelo de objetos del documento), el cual ofrece un acceso completo a todos los elementos, atributos y propiedades. El conjunto de eventos con los que responde SVG es amplio, algunos ejemplos son: *onmouseover* y *onclick*.

SVG es un lenguaje para contenido gráfico enriquecido. Algunas características de SVG en su especificación [?] son:

Tipo MIME⁴ de SVG - el tipo MIME de SVG es “*image/svg+xml*”[22], se recomienda que la extensión sea “.*svg*”, siendo todo en minúsculas.

Espacio de nombres (*namespace*) - el espacio de nombre asociado para SVG 1.1 es “*http://www.w3.org/2000/svg*”

Identificador público - para SVG 1.1 el identificador público es PUBLIC “*-//W3C//DTD SVG 1.1//EN*” y muestra el tipo de documento o DTD⁵ (W3C); el nombre del DTD (DTD SVG 1.1); y el lenguaje en el que el DTD está escrito (EN, para inglés).

Identificador de sistema - es una URI⁶ del actual DTD, para SVG 1.1 es “*http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd*”

⁵Document Type Definition

⁶Uniform Resource Identifier

La declaración del tipo de documento (*DOCTYPE*) especifica el tipo de elemento de la raíz del documento HTML, esto lo hace con el identificador público y un identificador del sistema. La declaración *DOCTYPE* le indica al verificador cual DTD debe aplicar al documento y probar si está escrito conforme al estándar.

Se presenta un ejemplo de la declaración de tipo de documento (DTD) para un documento SVG:

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

Compatibilidad de SVG con estándares

SVG llega a ser un excelente lenguaje para gráficos debido a que se integra de manera sencilla a otros estándares y especificaciones del W3C. En la siguiente lista se presentan algunos de las formas en que SVG es compatible:

- SVG es un aplicación de XML siendo compatible con la recomendación *eXtensible Markup Language (XML) 1.0* [23]
- SVG es compatible con la recomendación *Namespaces in XML* [24] que consiste en un método para calificar los nombres de elementos y atributos en documentos XML, asociándolos con el espacio de nombres identificado por un URI,
- los objetos gráficos SVG pueden ser estilizados por hojas de estilo CSS (especificación *Cascading Style Sheets Nivel 2*) [19] o XSL (versión 1.0 *XSL Transformations(XSLT)*) [25],
- SVG incluye el modelo de objetos del documento (DOM) conforme a la recomendación *Document Object Model (DOM) Level 1* [26]. Además, SVG - DOM incorpora las facilidades descritas en la recomendación *Document Object Model (DOM) Level 2* [27] que incluye el modelo de objetos CSS y manejo de eventos,
- Las características de las animaciones con SVG fueron definidas en colaboración con *W3C Synchronized Multimedia (SYMM) Working Group*, con los desarrolladores de la especificación 1.0 del lenguaje *Synchronized Multimedia Integration Language (SMIL)* [28].

A continuación se muestra un ejemplo de código para formar un rectángulo con SVG:

```
(1)<?xml version="1.0" standalone="no"?>
(2)<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
(3)  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
(4)<svg width="12cm" height="4cm" viewBox="0 0 1200 400"
(5)   xmlns="http://www.w3.org/2000/svg" version="1.1">
(6)  <desc>Ejemplo rect1 - Rectangulo</desc>
(7)  <rect x="1" y="1" width="1198" height="398"
(8)        fill="none" stroke="blue" stroke-width="2"/>
(9)  <rect x="400" y="100" width="400" height="200"
(10)        fill="yellow" stroke="navy" stroke-width="10" />
(11)</svg>
```

Descripción del código

En la línea (1) se define que el documento tiene formato XML y su versión, en la línea (2) y (3) se declara el tipo de documento (DTD), su identificador público y de sistema. En la línea (4) comienza la etiqueta del objeto SVG algunos atributos como la anchura y altura; la declaración del espacio de nombres en el atributo `xmlns` y la versión de SVG a utilizar en la línea (5). Una breve descripción del gráfico se coloca en la etiqueta `<desc>`. Los objetos gráficos, rectángulos en este caso, se definen en las líneas (7)-(10) con la etiqueta `<rect>` y los atributos que se definen son las coordenadas (x,y) en las que se comenzará a dibujar, la anchura y altura (`width`, `height`), el color de relleno (`fill`), el color de los bordes (`stroke`) y su ancho (`stroke-width`).

En específico, para los dispositivos móviles *iPodTouch/iPhone* es en septiembre de 2008 que se libera la actualización de su *Firmware 2.1* el cual incluye soporte a gráficos SVG dentro del navegador Web Safari en su versión para estos dispositivos móviles.

2.4. Introducción a la programación basada en *tags*

Después de la evolución en el desarrollo de servidores Web, API para usar CGI y servidores de páginas. Se tiene la intención de separar de forma clara las capas de desarrollo y de lógica de negocio⁷ [29]. Para lo anterior el desarrollo de *tags* JSP fue una primera propuesta de programación basada en *tags*; y así realizar la separación de código y la lógica de negocio. También está ColdFusion, que es un producto de *Allaire Corp.* que utiliza el concepto de desarrollo de sistemas Web con programación basada en *tags* y es utilizado ampliamente en el mercado. Un problema es su licencia propietaria, por lo que para varios desarrolladores es menos atractivo usarlo. En cambio, está la especificación de JSP que es abierta, siendo una solución puramente en ambiente de desarrollo Java se puede aprovechar de los beneficios de una plataforma cruzada, un amplio soporte y un lenguaje de programación funcional.

Beneficios de la programación basada en *tags*

A pesar de que la programación basada en *tags* JSP parece ser solamente una forma diferente de sintáxis que las JSP, trabajar con *tags* ofrece ventajas; por ejemplo, es muy cómodo para los desarrolladores de sistemas Web que utilizan HTML, quienes están muy familiarizados con el uso de *tags*. También, como característica a destacar es que varios *frameworks* para desarrollo de sistemas Web aceptan *tags*. Se pueden generalizar dos beneficios:

- La existencia de un estilo de página consistente y de fácil seguimiento,
- Los desarrolladores que dominen HTML pueden programar con *tags*, como se muestra en la Figura 2.6.

En su parte más fundamental, un *tag* es un grupo de caracteres que son analizados por un programa con el propósito de que éste realice una acción. En el caso de los *tags* HTML,

⁷Hágase referencia “lógica de negocio” al término en inglés *business logic*.

```
<%@ taglib prefix="tags" tagdir="/WEB-INF/tags" %>
<html>
  <head>
    <title>Ejemplo de Tags-JSP</title>
  </head>
  <body>
    <tags:geoqueries fid="1" type="limites" ></tags:geoqueries>
    <br/>
    <tags:geoqueries fid="1" type="hidrologia" ></tags:geoqueries>
  </body>
</html>
```

Figura 2.6: Código ejemplo de utilización de los *tags* JSP

el programa que lee los *tags* es el navegador Web, y las acciones que realiza van desde imprimir en pantalla palabras u objetos hasta crear formularios para reunir datos. Los *tags* JSP también son interpretados por un programa, pero a diferencia de HTML, los *tags* JSP son interpretados en el lado del servidor y no en el navegador Web o cliente. El programa que interpreta los *tags* es el motor de la aplicación del servidor en tiempo de ejecución (*TomCat* [30], *JRun* [31], *WebLogic* [32], entre otros). Cuando el motor de JSP detecta un *tag* propio (*Custom JSP tag*, en inglés), éste ejecuta código Java que ha sido definido previamente en el *tag*. Las tareas más comunes asociadas a los *tags* JSP incluyen obtener información de una base de datos, el formato de texto y el regreso de código HTML al navegador Web. Al trabajar con *tags* se reducen o eliminan los *scriptlets* de una JSP de las siguientes formas [33]:

- Un *tag* no es más que un componente Java que toma sus argumentos como atributos del *tag* y un cuerpo. No es necesario al inicializar algún código Java,
- Un JSP requiere una cantidad considerable de *scriptlets* para tareas como iteraciones, establecimiento de valores iniciales y generar alguna parte de código HTML que es esté condicionada. En cambio, todas las tareas mencionadas pueden ser abstraídas en pocos *tags*,
- En varios casos, un componente *JavaBean* es configurado y activado usando *scriptlets*. Se pueden desarrollar un conjunto de *tags* JSP para que realicen estas tareas sin ningún código Java,
- Los *tags* pueden crearse para diversas operaciones, tales como enviar correos electrónicos y conectar a bases de datos, lo que reduce el número de *scriptlets* necesarios dentro de una JSP.

Resumen del capítulo

Se comenzó con una breve historia de como la *World Wide Web* ha impactado en la creación de los sistemas de información geográfica y que empresas han invertido en dichos sistemas. Además, se mencionaron los proyectos similares a mi trabajo de tesis, los cuales se describen a detalle en el capítulo 3 de este documento. También, se definen las características de los sistemas de información geográfica, sus aplicaciones potenciales de los SIG

y esencialmente los subsistemas que componen un SIG, que son: una entrada de datos que reúne y preprocesa datos espaciales; el subsistema de almacenamiento y consultas; el subsistema de manipulación y análisis; y el subsistema de visualización para desplegar los datos. También, se realiza una interesante comparación, subsistema a subsistema, del proceso de un sistema de información geográfica *vs* el proceso de la cartografía tradicional, obteniendo que los SIG tienen la ventaja de analizar los datos espaciales más rápido, debido a que usan el potencial de las computadoras para procesar y almacenar los datos digitalmente, los cuales se pueden consultar mediante lenguaje SQL. Se introdujo a las tecnologías sobre paradigma Web 2.0 y AJAX. Se presentaron los formatos gráficos Canvas y SVG con ejemplos de estructuras de código para generar gráficos dinámicamente con ayuda de *tags* JSP, a los cuales también se introdujo, mencionando sus ventajas.

En el próximo capítulo se definen los conceptos básicos para comprender que es un sistema de bases de datos espaciales y que es un sistema de información geográfica. Se presentan los tipos de datos espaciales y la especificación que siguen estos sistemas. Además de una breve introducción a *PostGIS* para entender como se respalda y se crea una base de datos espacial en *PostgreSQL*.

Capítulo 3

Sistemas de bases de datos espaciales

3.1. Definición y clasificación de los datos espaciales

Los datos espaciales son también comúnmente llamados “datos georeferenciados geográficamente” y “datos geoespaciales” [34]. Los datos espaciales son datos que pueden ser visualizados, manipulados y analizados por sus atributos, que denotan una localización sobre la superficie del globo terraqueo. Estos atributos espaciales son normalmente dados en forma de coordenadas, que dan la posición y la estructura de una característica representada geográficamente. Los datos espaciales tienen dos propiedades importantes:

- se referencia a un “espacio geográfico”, es decir, son datos registrados y aceptados sobre un sistema de coordenadas geográfico para representar áreas de la superficie terrestre. Por lo tanto, si se tienen diversas fuentes, se puede hacer referencias cruzadas e integrarlas espacialmente.
- son representados en una variedad de escalas y cuando son almacenados en escalas relativamente pequeñas, representan grandes áreas de la superficie terrestre y deben ser generalizados y simbolizados.

La unidad básica de los datos espaciales en la representación vectorial es llamada “objeto geográfico”, el cual se identifica como una característica del mundo real o fenómeno representado por un punto, línea o polígono. En las bases de datos espaciales, los datos vectoriales pueden ser almacenados como parte de la base topográfica, cuya función es proveer la referencia espacial a sistemas para su visualización, manipulación y análisis.

Las características que comúnmente se encuentran en una carta topográfica son: vías de transportación, ríos, áreas urbanas y características de vegetación.

Los datos espaciales vectoriales pueden describir el estado de un ambiente natural (*e.g.* áreas con problemas de contaminación, bosques y la calidad del aire y agua) así como también, datos acerca de las actividades de los humanos o la utilización de la tierra y sus recursos (*e.g.* registro de tierra, uso y división de tierra y redes de transporte) [34].

Los datos espaciales incluyen un componente que describe la localización espacial de un objeto geográfico y un componente de atributos para describir sus propiedades y toponimia. El componente espacial puede ser representado mediante las siguientes formas:

- Representación *raster*.- este modelo usualmente se aplica cuando se tienen mallas de puntos sobre campos continuos, como la elevación, temperatura y concentraciones químicas; y cuando se utilizan mallas de celdas, representan imágenes; se basa en almacenar matrices de valores. Si las celdas se categorizan por números, uno o más atributos pueden ser asignados a la celda. La unidad de área en las mallas de celdas se llaman pixeles.
El modelo *raster* fue diseñado para el análisis, modelado y procesamiento de imágenes.
- Representación vectorial.- este modelo se basa en la representación arco-nodo y es utilizado para representar áreas, líneas y puntos. Los arcos pueden formar conjuntos de líneas (*e.g.* carreteras o ríos). Además, mediante arcos se forman polígonos para representar áreas (*e.g.* lagos o bosques). Un caso especial del modelado de datos vectorial es el modelado de datos puntual, que es una colección de puntos independientes que, establecidas las coordenadas, representan lugares puntuales (*e.g.* una escuela o una iglesia) que se distribuyen irregularmente.

3.2. Estructura de los datos espaciales

3.2.1. El concepto de un dato espacial tipo *Geometry*

El campo de la geometría, se entiende comúnmente como la rama de las matemáticas que estudia las propiedades y relaciones de puntos, líneas, ángulos, superficies y sólidos en cualquier dimensión [35]. En el contexto del procesamiento de datos espaciales, la palabra *geometry* supone un nuevo significado cuando el OGC (*Open Geospatial Consortium*) formalizó su uso en la publicación *OpenGIS Simple Feature Specification for SQL* [35]. En este comunicado el OGC propuso una jerarquía de tipos de datos, llamada *geometry object model*, la cual permite que características espaciales puedan ser representadas en una base de datos. En el modelo, la palabra *geometry* es usada para representar una característica espacial como un “objeto” que tiene al menos un atributo *geometric type* (según la especificación) en la base de datos.

La clase raíz del modelo jerárquico *geometry object model*, es la clase *geometry*, de la cual no se pueden crear instancias. La superclase *geometry* tiene cuatro subclases llamadas: *point*, *curve*, *surface* y *geometry collection*. Se pueden crear instancias de estas subclases las cuales contienen un conjunto de métodos, que son utilizados para verificar sus respectivas propiedades geométricas, y soportar el análisis espacial [36]. Existen varios tipos *geometry* en el modelo, como se muestra en la Figura 3.1, en la cual, los nombres de los objetos se mantienen en inglés para seguir la especificación y mantener estándar [35]. Estos tipos son primitivas gráficas utilizadas para construir la geometría de objetos geográficos que usan una o más primitivas. Un poste de luz, un segmento de carretera, una parcela son ejemplos de estas primitivas gráficas. Estos objetos son llamados *simple geometries*. Un grupo de islas que es tratado como una sola entidad geográfica, *e.g.* Nueva Zelanda y Fiji, son construidas con múltiples primitivas gráficas. Estos objetos son llamados *complex geometries* [36].

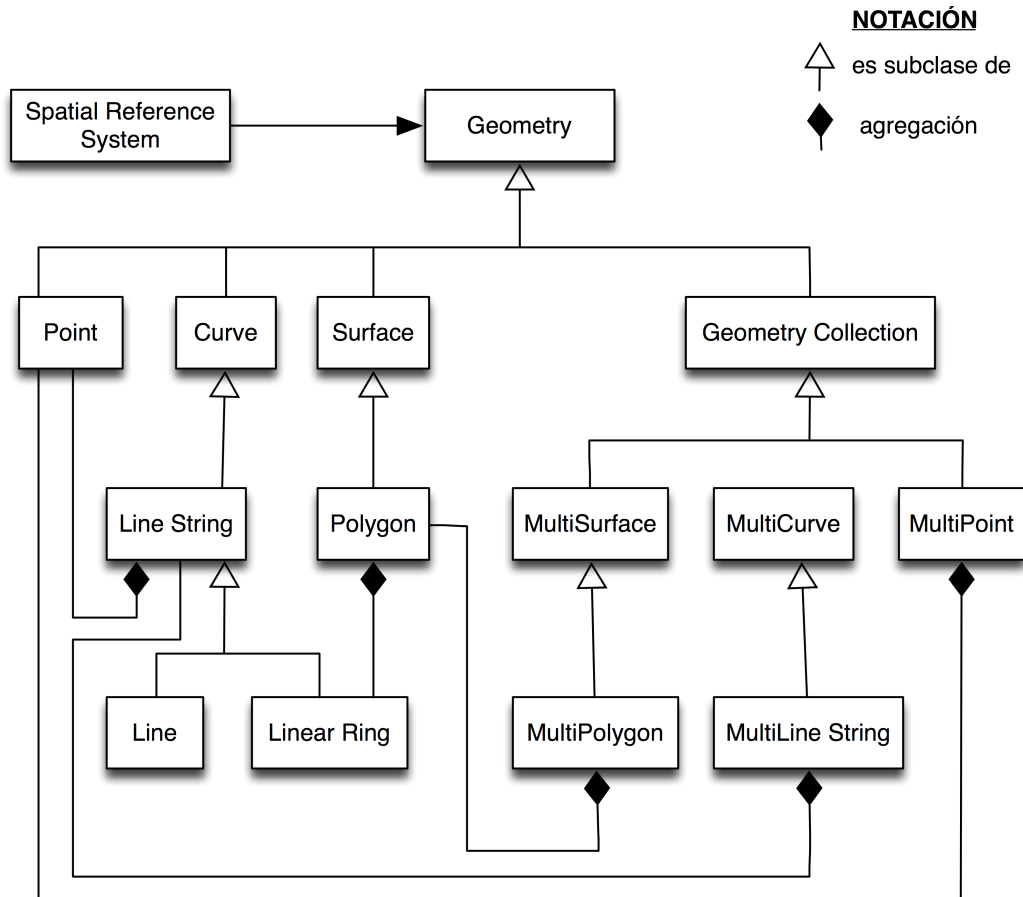


Figura 3.1: El modelo de la OGC (*Geometry object model*)

Las primitivas gráficas son espacialmente referenciadas a un sistema de coordenadas, la posición y orientación de un objeto siempre están relacionados a un espacio geográfico particular. Esta referencia espacial al sistema de coordenadas ayuda a unir los objetos geográficos a otros, no importando las primitivas gráficas con las cuales fue construido. Además de su localización, un objeto tipo *geometry* tiene otros atributos que describen sus características, tales como el nombre, clasificación y dimensión.

En una base de datos espacial, los objetos que tienen los mismos atributos forman una capa (*layer*, en inglés) o también llamada *feature class*; que consecuentemente son almacenadas en la misma tabla. En la Figura 3.2 se muestran las relaciones entre objetos *geometry* y otros elementos utilizados para construir y representar características espaciales en una base de datos.

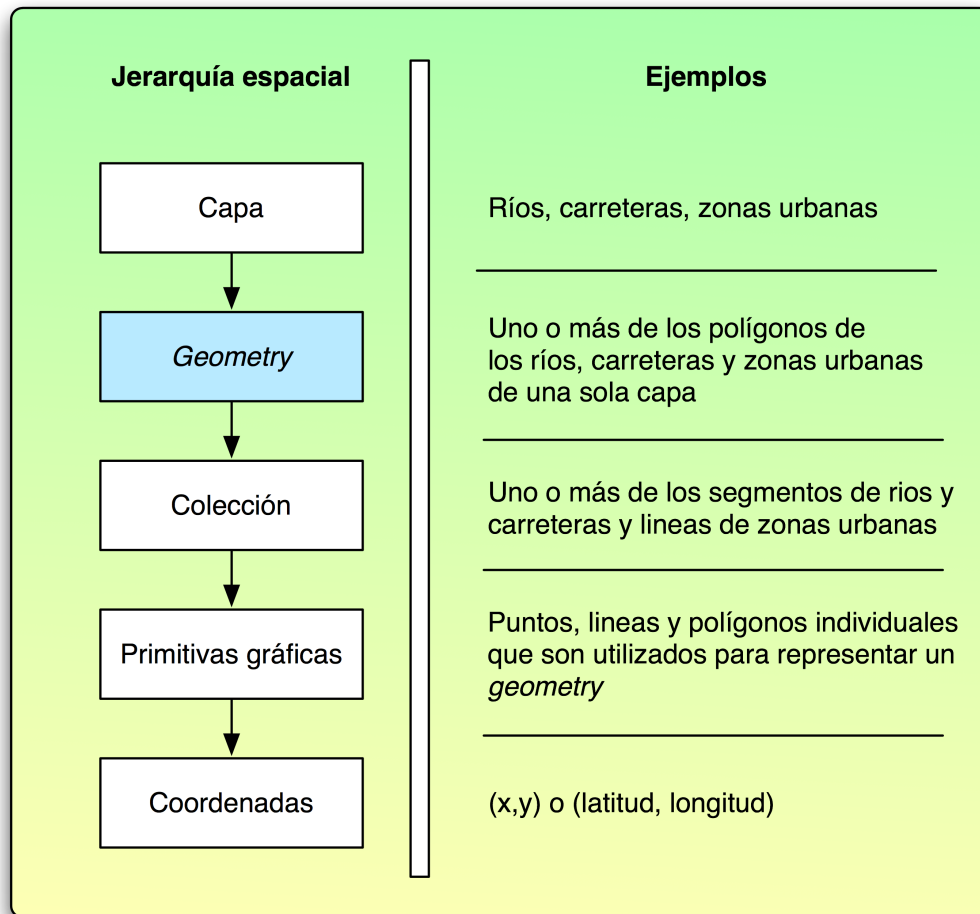


Figura 3.2: La relación del concepto *Geometry* con otros elementos

El concepto de topología y estructuras de datos topológicas

La topología es el campo de las matemáticas que estudia la propiedades de las figuras geométricas que permanecen sin cambio, cuando la forma de la figura es doblada, aplastada, o distorsionada sin romperse [37]. Cuando la topología es aplicada a estructuras de datos espaciales, ésta típicamente define relaciones espaciales entre objetos vecinos, incluyendo su adyacencia, conectividad y sobre el contenido.

Una estructura de datos topológica es aquella donde las relaciones espaciales entre características del mundo real pueden ser almacenadas explícitamente. En las estructuras de datos topológicas, un nodo (*point*) es una característica espacial de 0 dimensiones en donde uno o más arcos (*arcs*, *chains* y *segments* o *lines*) se conecta para formar una unión topológica *junction*. Un arco es una característica espacial de 1 dimensión que es formado por una línea dirigida desde un nodo a otro. Un polígono (también llamado *topological ring*, *face* o *area*) es de 2 dimensiones y a través de una secuencia de arcos se cierra por sí mismo [38].

En la Figura 3.3-a los polígonos son representados por una secuencia de coordenadas que son usadas para dibujarlos gráficamente. Un conjunto de datos espaciales representado de esta manera se dice que tiene una estructura “polígono completo” (*full polygon*). Una estructura de “polígono completo” es también conocida como “estructura de datos cartográfica” (*cartographic data structure*) por que los datos de un mapa digital en el ambiente de la cartografía asistida por computadora son típicamente almacenados de esta forma [36].

Cuando la topología se divide en dos polígonos, los límites de los polígonos en la Figura 3.3-b son separados en los puntos de intersección “a” y “b” para formar los arcos 1, 2 y 3. Al mismo tiempo, las relaciones espaciales son creadas y almacenadas explícitamente en dos tablas de topología. La tabla **Topología de la red** almacena las relaciones espaciales de conectividad de los arcos definidos por nodos de origen (*nodo_o*) a los nodos destino (*nodo_d*). Por otra parte, la tabla de **Topología de polígonos** almacena las relaciones espaciales de adyacencia de los polígonos definidos a la izquierda de los arcos (*poligono_izq* y *poligono_der*). Un conjunto de datos espaciales son representados con relaciones espaciales almacenados explícitamente como una estructura de datos topológica.

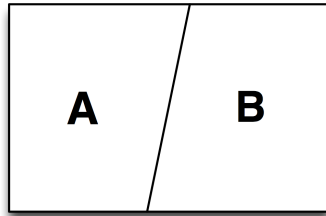
Estructuras de datos no topológicas

El modelo de datos *shapefile* fue introducido con la liberación de ArcView 2.0¹[39] por el Instituto de investigación de sistemas ambientales (ESRI[40], por *Enviromental Systems Research Institute, Inc*) a principios de la década de 1990, el cual contribuyó al uso de estructuras de datos con “polígonos completos” para el mundo de los datos espaciales, una estructura de este tipo almacena polígonos individualmente, como entidades discretas muy parecidas a la estructura de datos cartográfica, y es una estructura de datos no topológica por que no almacena explícitamente relaciones espaciales. Existen diversas aplicaciones que no requieren el uso de relaciones topológicas.

Sin embargo, como lo notó Strand [41], los datos espaciales que siguen el formato *shapefile* tienen algunas limitaciones que detienen su uso en algunas aplicaciones, *e.g.* incompatibilidad con los principios y técnicas de administración de bases de datos relacionales, ausencia de topología para un análisis espacial eficiente y el poco soporte para transferir metadatos que describan los datos espaciales y sus características.

Ambas estructuras, topológica y no topológica o cartográfica tienen su lugar en el diseño y desarrollo de sistemas de información espaciales, por que diferentes usuarios tienen diferentes requerimientos para el uso de relaciones topológicas.

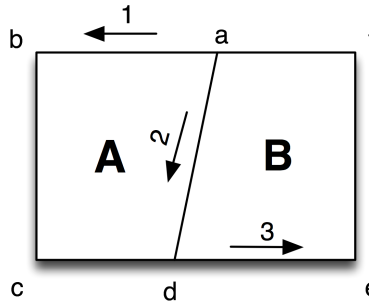
¹Software para visualizar, administrar, crear y analizar datos espaciales en SIG.



Poligono A = (403600, 275700), (403000, 275700), (403000, 275000), (403300, 275000), (403600, 275700)

Poligono B = (403600, 275700), (403300, 275700), (404000, 275700), (404000, 275000), (403300, 275000), (403600, 275700)

(a) Estructura de datos cartográfica (no topológica)



Poligonos	
Id_poligono	Arcos
A	1,2
B	2,3

Arcos	
Id_arco	Vertices
1	b,c
2	-
3	e,f

Coordenadas		
Id_vertice	X	Y
b	403000	275700
c	403000	275000
e	404000	275000
f	404000	275700

Nodos		
Id_nodo	X	Y
a	403600	275700
d	403300	275000

Topología de la red		
Id_arco	nodo_o	nodo_d
1	a	d
2	d	a
3	d	a

Topología de polígonos		
Id_arco	poligono_izq	poligono_der
1	A	Universo
2	A	B
3	B	Universo

(b) Estructura de datos topológica

Figura 3.3: Estructuras de datos topológica y cartográfica

3.2.2. Definición de sistemas de bases de datos espaciales

Los sistemas de bases de datos espaciales son definidos como una clase de sistemas de base de datos que cumplen con las siguientes características [42]:

- Un sistema de bases de datos espaciales es un sistema de base de datos,
- ofrece tipos de datos espaciales (SDT, por *spatial data types*), un modelo de datos y un lenguaje de consulta,
- soporta tipos de datos espaciales en su desarrollo, dando un indexado espacial y algoritmos eficientes para la operación reunión espacial (*spatial join*).

La primera característica enfatiza el hecho que estos sistemas son completamente sistemas de bases de datos, capaces de soportar todos los estándares de modelado y consultas, pero con la funcionalidad adicional de efectuar tareas específicas a datos espaciales. Esta habilidad de manipular y relacionar datos espaciales y no espaciales distingue a sistemas de bases de datos espaciales de otros sistemas que usan datos con referencias geográficas, tales como sistemas cartográficos, diseño asistido por computadora y sistemas de procesamiento de imágenes para el sensado remoto [36].

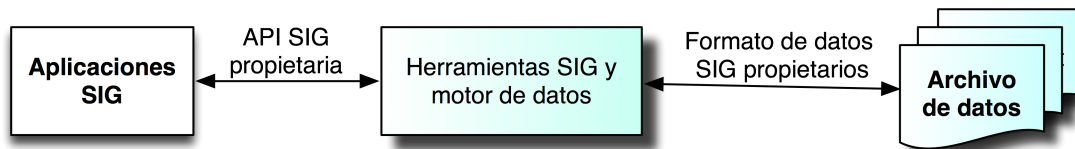
Los tipos de datos espaciales, segunda característica, se refieren al uso de puntos, líneas y polígonos como forma de representación geométrica de objetos o entidades en el espacio; como también sus relaciones (*e.g.* i es adyacente a j), propiedades (*e.g.* $\text{area}(a) > 1000 \text{ km}^2$) y operaciones (*e.g.* objetos m que se traslapan con el objetos n) [42].

Los sistemas de bases de datos modernos se caracterizan porque pueden manipular grandes cantidades de datos, estos sistemas deben conservar un acceso eficiente mediante técnicas de indexado [43]. Esta característica también es necesaria en los sistemas de bases de datos espaciales. Sin embargo, los datos en bases de datos espaciales son representados y referenciados en dos y tres dimensiones, así los datos en una base de datos espacial deben ser indexados espacialmente. Lo que permite que los datos sean accedidos y analizados por localización (utilizando las coordenadas geográficas) y por las relaciones topológicas que definen la posición de objetos en el espacio relativo a otro (adyacencia, contenido y conectividad). Juntos, la localización y la topología permiten varios métodos de traslape y combinación espacial de objetos en las capas de mapas, este tipo de consultas se llaman reuniones espaciales, y son usadas para propósitos de análisis.

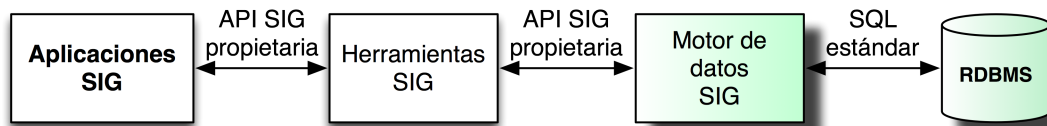
La Figura 3.4 muestra la evolución de los conceptos de procesamiento de datos espaciales y técnicas para el desarrollo de sistemas de información geográfica (SIG) desde principios de las década de 1970 cuando la primera generación de ellos fueron desarrollados [36]. Hasta mediados de la década de 1990, el procesamiento de los datos espaciales fue caracterizado por una dependencia de software SIG propietario, API y formatos de datos propietarios (véase Figura 3.4-a). La necesidad de trabajar con grandes cantidades de datos espaciales y los avances en la tecnología de las bases de datos llevaron a el incremento de sistemas administradores de bases de datos relacionales (RDBMS, por sus siglas en inglés) para aplicaciones SIG. Prácticamente todo el procesamiento de los datos espaciales era todavía efectuado dentro del SIG (véase Figura 3.4-b).

Con la llegada de los sistemas administradores de bases de datos objeto-relacional (ORDBMS, por sus siglas en inglés) a finales de la década de 1990 [36] fue posible utilizar

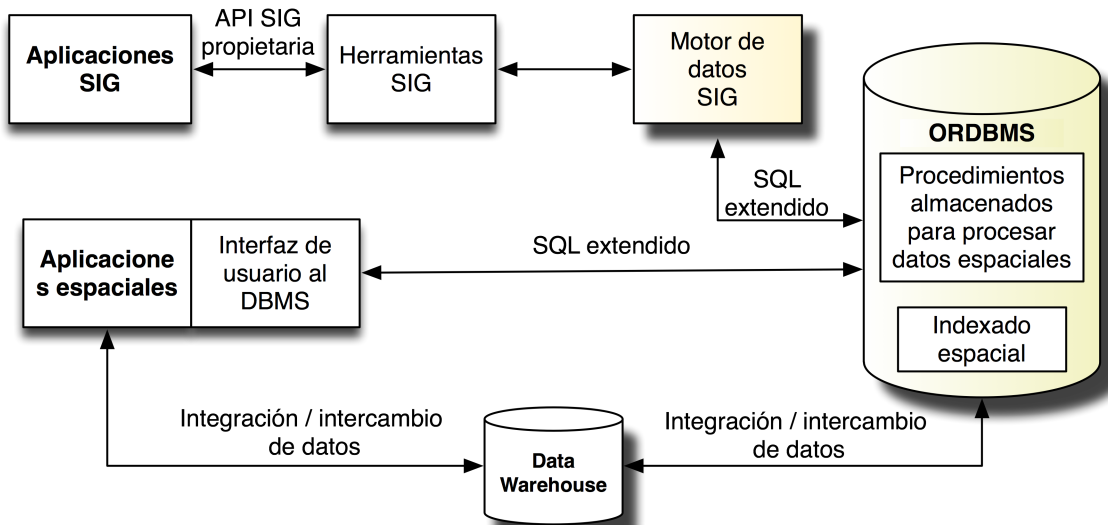
la funcionalidad de los DBMS para procesar y aplicar los datos espaciales para crear aplicaciones. Utilizando la capacidad del lenguaje SQL extendido de manejar datos espaciales, un ORDBMS puede manejar datos espaciales y no sólo datos de atributos. Un ORDBMS puede ser conectado a otros sistemas de bases de datos relativamente fácil dentro de una red local o a través de la Internet, creando una parte integral de la arquitectura de cómputo de organizaciones, ya sean privadas, gubernamentales o de la educación (véase Figura 3.4-c).



(a) Procesamiento de datos espaciales basado en archivos utilizando un SIG a mediados de los 90's



(b) Procesamiento de datos espaciales basado en DBMS utilizando un SIG a finales de los 90's



(c) Ambiente de procesamiento de datos espaciales actual

Figura 3.4: Evolución de los conceptos de procesamiento de datos espaciales

3.3. Relación entre bases de datos espaciales y SIG

En ambientes de procesamiento de datos espaciales existe una división entre sistemas de bases de datos espaciales y SIG (véase Tabla 3.1). Tomando la ventaja de los avances en las tecnologías computacionales, los sistemas de bases de datos son desarrollados para almacenar y gestionar grandes cantidades de información referenciada geográficamente basada en su localización. Estos sistemas indexan los datos espacialmente para crear consultas eficientes, proveer mecanismos y procedimientos requeridos para proteger los datos de una destrucción física y pérdida o degradación de la integridad lógica. Sin embargo, los sistemas de bases de datos espaciales son generalmente débiles en su funcionalidad para la edición de datos, en el análisis de datos y generación de mapas y otros productos información cartográfica. Estos procesos y funciones son manejados de mejor forma por SIG, los cuales son primordialmente diseñados para reunir y usar los datos espaciales para esos propósitos.

<i>Sistemas</i>	<i>Tareas primarias</i>
Sistemas de información geográfica (SIG)	<ul style="list-style-type: none"> ● Reunir y editar datos ● Análisis de datos ● Generación de mapas y productos de información cartográfica
Sistemas de bases de datos espaciales	<ul style="list-style-type: none"> ● Almacenamiento y gestión de datos ● Indexado espacial ● Seguridad e integridad de los datos ● Consultas sobre datos espaciales

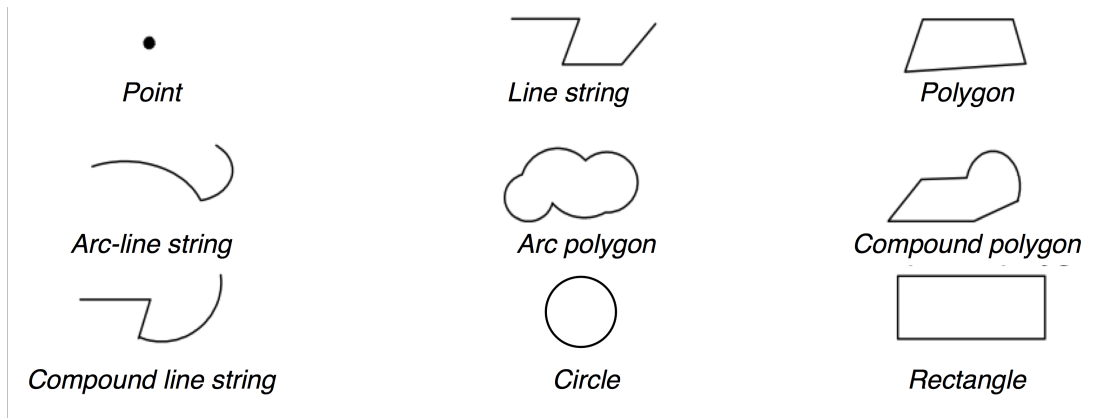
Tabla 3.1: La división de trabajo entre sistemas de bases de datos espaciales y SIG

Tipos de datos espaciales

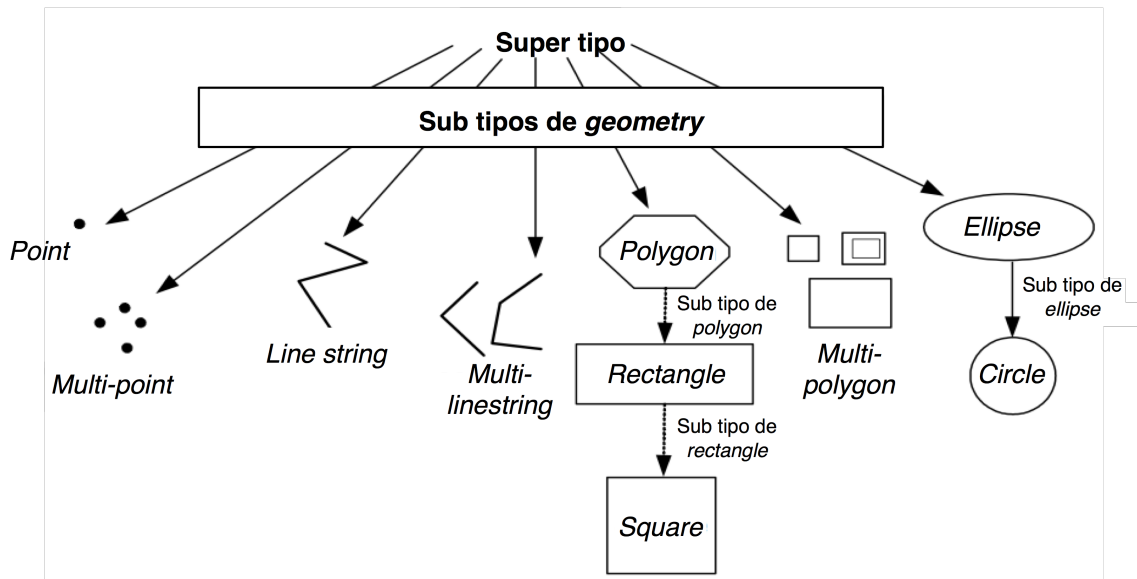
Convencionalmente, los sistemas de bases de datos fueron diseñados para gestionar y procesar datos alfanuméricos representados por cadenas de caracteres, valores numéricos, fechas y expresiones lógicas o booleanos [43]. No existía un soporte en los DBMS convencionales para almacenar y procesar datos espaciales representados por puntos, líneas y polígonos. Entonces, varios productores de software para bases de datos definieron esa capacidad para los tipos de datos espaciales (SDT), que incluye gestionarlos y procesarlos. Aunque el *Open Geospatial Consortium* (OGC) ofrece el modelo *geometry object model* para estandarizar los tipos de SDT, los productores de software para bases de datos desarrollaron el concepto de SDT de diferentes maneras.

El software *Oracle Spatial* [44], por ejemplo, tiene nueve tipos de datos espaciales llamados *geometric primitive types* que incluyen *points* y *point clusters*, *line strings*, *polygons*, *arc strings*, *arc polygons*, *compound polygons*, *compound line string*, *circle* y *optimised rectangles* (véase Figura 3.5-a). Por otra parte está *DB2 Spatial Extender* de IBM [45], que utiliza el término *geometry type* para describir los SDT, y están organizados en una jerarquía de subtipos de geometría, incluyendo *points*, *multi-points*, *line strings*, *multi-line*

strings, *polygons*, *multi-polygons* y *ellipses* (véase Figura 3.5-b). También está *ArcGIS geodatabase* de ESRI [46], en donde los objetos espaciales son representados por cuatro tipos de SDT que describen su forma, llamados *point*, *multipoint*, *polyline*, *polygon* (véase Figura 3.6-a). Otra opción es *PostGIS* [47], que es una extensión al sistema de bases de datos objeto relacional *PostgreSQL* [48], que se libera en su versión 1.0 en el 2001, los SDT que soporta incluyen *points*, *line strings*, *polygons*, *Multi points*, *Multi line strings*, *Multi polygons* y *geometry collections* (véase Figura 3.6-b).



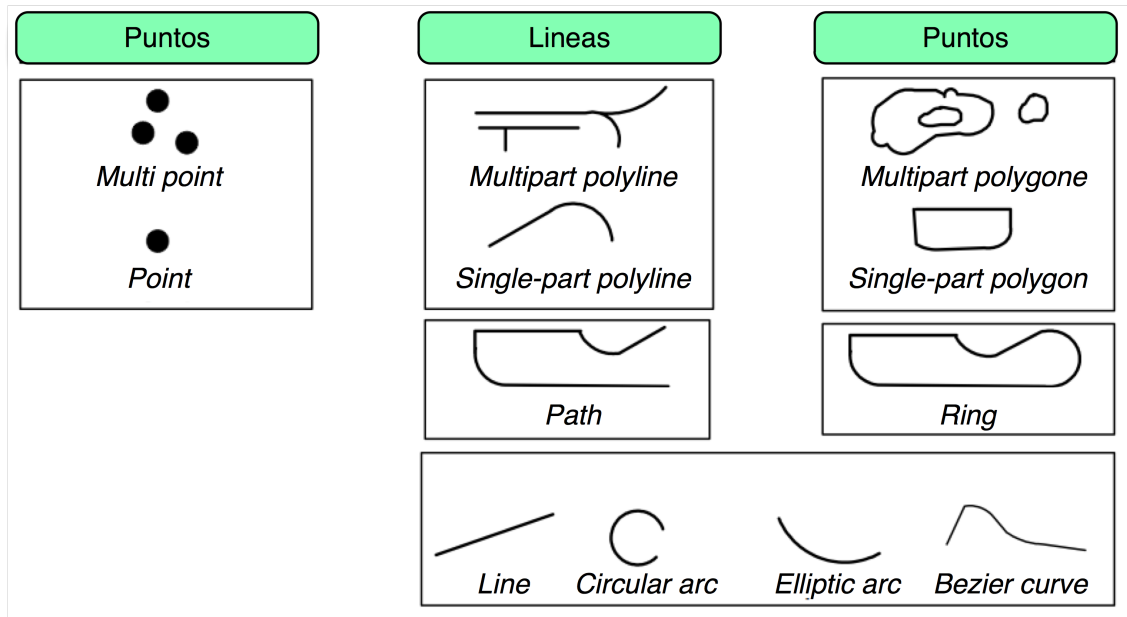
(a) Tipos *Geometry* en el modelo orientado a objetos de Oracle Spatial



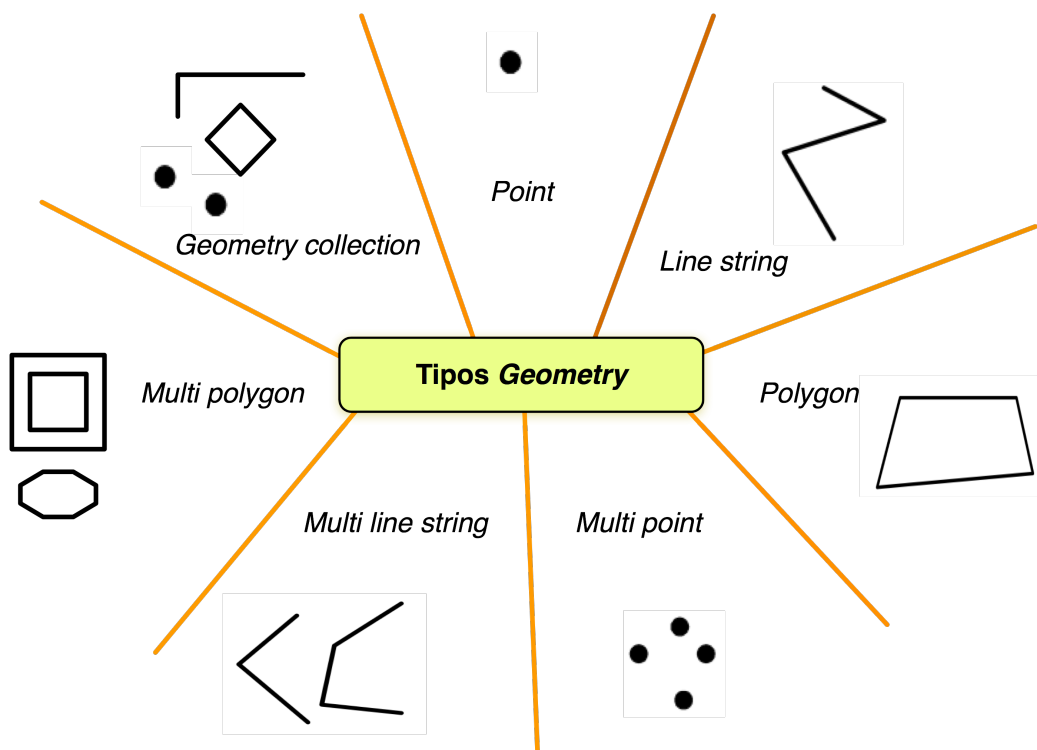
(b) Tipos y subtipos *Geometry* de DB2 Spatial Extender

Figura 3.5: Tipos de datos espaciales *Geometry* en *Oracle Spatial* y *DB2 Spatial Extender*

Además de datos espaciales vectoriales como puntos, líneas, polígonos y superficies, es posible definir datos espaciales *raster* como un tipo de dato abstracto (ADT, en inglés) en bases de datos objeto-relacionales. Como resultado, los datos espaciales vectoriales y *raster* pueden ser almacenados y gestionados dentro de un mismo sistema de bases de datos espacial.



(a) Tipos *Geometry* de ArcGIS Geodatabase



(b) Tipos *Geometry* de PostGIS

Figura 3.6: Tipos de datos espaciales *Geometry* en ESRI Geodatabase y PostGIS

3.4. Introducción a *PostGIS*

PostGIS [47] es un software de código abierto (*opensource*), disponible de forma gratuita, el cual se rige por los estándares de el OGC [35]. Es una extensión al sistema administrador de bases de datos *PostgreSQL* [48]. A *grosso* modo, *PostGIS* añade funciones espaciales como obtener la distancia, área y tipos de datos geométricos de la base de datos. *PostGIS* es muy similar a *ArcGIS geodatabase*, *Oracle Spatial* y *DB2 Spatial Extender*. La última versión viene en el paquete de instalación de *PostgreSQL* como una opción.

Los comandos mostrados a continuación son importantes por que al ser datos espaciales la creación de una base de datos en *PostgreSQL / PostGIS* no es como comúnmente se realiza. Las instrucciones se ejecutaron en el sistema operativo *MacOS 10.5.4, PostgreSQL 8.0.6, PostGIS 1.3.1*

Respaldo y recuperación de una base de datos espacial

```
$ pg_dump -Fc -h direccioIP_remota -p 5432 -U usuario -d  
bd_espacial_fuente > respaldo_bd_espacial.dump
```

```
$ createdb nueva_bd_espacial  
$ createlang plpgsql nueva_bd_espacial  
$ psql -d nueva_bd_espacial -f lwpostgis.sql  
$ psql -d nueva_bd_espacial -f spatial_ref_sys.sql  
$ pg_restore -Fc -d nueva_bd_espacial -U usuario -W  
respaldo_bd_espacial.dump
```

Importación de archivos shapefile a PostGIS

```
$ shp2pgsql -s > salida.sql  
psql -h servidor -d base_datos -U usuario -f salida.sql
```

Ventajas de *PostGIS*

En términos de sistemas de bases de datos espaciales, *PostGIS* es una extensión para gestionar bases de datos espaciales de código abierto. Mientras MySQL tiene algunas capacidades para manejar datos espaciales, pero estas son extremadamente limitadas particularmente en la selectividad de las funciones relacionales espaciales, también en la capacidad de crear el indexado espacial: Otra carencia de MySQL es la débil adecuación de algunas funciones especificadas por el OGC como *intersection* y *buffer*.

Cuando *PostGIS* se compara con sistemas de bases de datos comerciales, *PostGIS* tiene más funciones de las que tienen sistemas comerciales como *Oracle Spatial* [44], *DB2 Spatial Extender* [45], *Informix Spatial Blade* [49]. Un argumento para sustentar lo anterior es, por ejemplo, *Oracle Locator* [44] tiene un límite en el conjunto de funciones espaciales. *Oracle Locator* no tiene algunas funciones de manipulación geométrica y análisis espacial, como *centroid*, *buffering*, *intersection* y funciones de agregación espacial. Aunque *Oracle*

Spatial tiene la funcionalidad de soportar el sistema de coordenadas *geodetic*², mientras *PostGIS* no lo soporta. Por otra parte, *Oracle Spatial* y *Oracle Enterprise* [50] se elevan hasta un costo de \$60, 000 por procesador.

Una característica de *SQL Server 2008* [51] es el soporte del sistema *Geodetic*; aunque pierde un poco de vista la perspectiva actual sobre el desarrollo de sistemas de bases de datos espaciales, ya que no contempla la habilidad para transformar de sistema de referencia espacial a otro directamente en la base de datos; trabaja sólo con el sistema operativo *Windows* y no maneja opciones pensando en un ambiente *Unix* o una plataforma cruzada (*cross-platform*). *SQL Server 2008* tal vez sea el más cercano a *PostGIS* en términos de funcionalidad y precio. Se planea que en futuras versiones de *PostGIS* se añada el soporte *Geodetic* y soporte a redes topológicas con el proyecto *PgRouting* [52].

Debido al gran uso de aplicaciones para ver mapas y analizar datos espaciales, las personas en el mundo buscan SIG a los cuales acceder desde cualquier lugar [5]. Y es por eso, que la visualización de mapas y la industria de análisis de datos espaciales dejará de ser sólo para especialistas. También, han crecido de gran manera las aplicaciones Web llamadas *mashups* que mezclan la API de mapas de *Google*, *Microsoft Virtual Earth*, entre otros; con aplicaciones Web dando como resultado nuevas formas de compartir mapas y tener conocimiento de objetos geográficos.

Entre los objetivos de *PostGIS* es ofrecer funcionalidad para aplicaciones que estudien la destrucción debido a huracanes, la migración de aves, seguimiento y rastreo por GPS, definir límites y densidades basados en factores socio-ecológicos, en donde se necesita almacenar una gran cantidad de datos; de atributos normales, como espaciales, para ofrecer un sistema en el cual se pueda consultar de forma fácil, que es lo que ofrecen los sistemas gestores de bases de datos relacionales (RDBMS), como *PostgreSQL*. Este es el nivel en donde *PostGIS* y otros sistemas de bases de datos espaciales son realmente útiles.

Resumen del capítulo

Este capítulo tuvo el propósito de introducir a los sistemas de bases de datos espaciales, debido a que la API para visualización de datos espaciales, que es el objetivo de este trabajo de tesis, y los desarrolladores que deseen utilizar la API deben comprender los datos sobre los cuales se efectuarán las consultas. Para la comprensión de los datos, se definió en la primera parte del capítulo que es un dato espacial y su clasificación. Después, se presentó la estructura que siguen los datos espaciales, basándome en la especificación del OGC (*Open GIS Consortium*) [35]. Entonces, se definió que un sistema de base datos espaciales es un sistema de base de datos que ofrece la capacidad para almacenar tipos de datos geométricos, y un lenguaje de consulta. También, se remarca la diferencia entre un SIG y un sistema de bases de datos espaciales. Por último, se introdujo brevemente al uso de *PostGIS* para respaldar y recuperar una base de datos espacial.

En el próximo capítulo se presentan los proyectos similares al que se realizó en esta tesis.

²*Geodetic* se refiere al sistema de coordenadas **grs80**.

Primero se describen las API existentes para generar SIG en Web, ya sea para ambiente Web o para PC. Después se presentan SIG en Web para comparar los SIG creados para validar la biblioteca de *tags* para tener un parámetro con que comparar el subsistema de visualización de mapas en dispositivos móviles.

Capítulo 4

Estado del arte

En la búsqueda de los principales resultados en el área de investigación, se encuentran dos grandes vertientes: los sistemas Web de información geográfica y las aplicaciones Web 2.0 para dispositivos móviles. A continuación se describen los resultados de las recientes investigaciones en estas áreas.

4.1. API para el desarrollo de sistemas de información geográfica

4.1.1. *Mapaccel*

Mapaccel [6] es un *framework* para desarrollar sistemas de información geográfica, los cuales se basan en el servidor de mapas *mapaccel IMS* creado por *Territorium Online* (el cual está implementado usando UMN *MapServer* [53]) o ArcIMS creado por ESRI. Ofrece soluciones que permiten crear aplicaciones Web complejas para consultar o editar datos geográficos o espaciales. Lo anterior sin ser un desarrollador de software.

El bloque principal de *Mapaccel* es el *GeoApplicationServer* (GAS), el cual está compuesto por más de 200 clases en lenguaje java. El servidor controla el acceso al servidor de mapas y ofrece las funciones para los servicios que se presentan en la aplicación Web. *mapaccel* corresponde a un arquitectura orientada a servicios (SOA) [54]. Algunas características importantes de *Mapaccel* son:

- La utilización del sistema llamado *mapsnap*, que permite configurar y crear SIG en Web de cualquier complejidad de manera fácil y rápida,
- *mapaccel* la producción de documentos HTML ligeros, debido a que todas las peticiones a la aplicación Web son manejadas por el servidor,
- la digitalización en línea de puntos, líneas, polígonos y atributos,
- la integración con DBMS relacionales,
- la disponibilidad de un editor de mapas (*mapedit*),

- la presencia en el servidor únicamente de código en Java,
- la fácil administración de los estilos de las aplicaciones Web a través del soporte de CSS,
- el soporte con *mapaccel IMS* de las siguientes fuentes de datos espaciales: OGC capas de WMS¹, capas de WFS, capas de *PostGIS*, capas de *Oracle Spatial*, capas de *ArcGIS Geodatabase*, *shapefiles* y *geoimages*,
- el soporte para la mayoría de navegadores Web.

En la Figura 4.1 se presenta el esquema general de los componentes del *framework mapaccel*, en la parte izquierda de la figura se presenta el servidor de mapas (*Mapaccel IMS*), en la parte central se muestra el servidor de aplicaciones geográficas (*mapaccel GAS*) y en la parte derecha se indican los dos posibles usos del *framework*: aplicaciones de escritorio o sistemas Web de información geográfica.

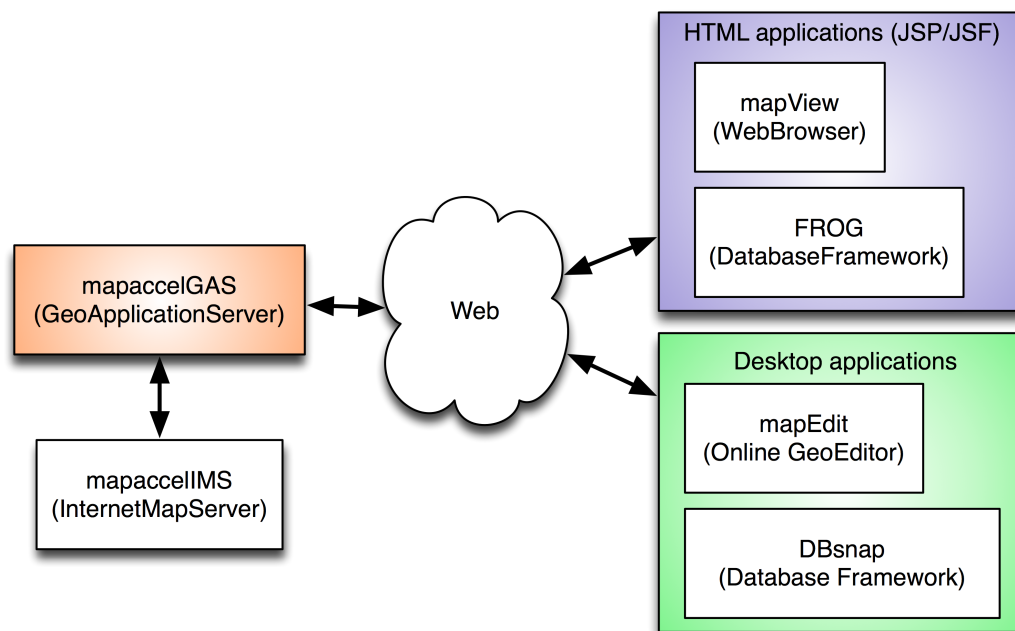


Figura 4.1: Arquitectura general del framework *mapaccel*

Mapedit es el cliente de edición avanzada de datos espaciales que son publicados posteriormente en Web por los servicios de las aplicaciones. *Mapedit* soporta la edición en línea con las siguientes combinaciones en el servidor:

- *mapaccel IMS/PostGIS*
- *mapaccel IMS/Oracle Spatial*

¹ *Web Mapping Service*, en inglés

- *mapaccel IMS/ArcGIS Geodatabase*
- *ArcIMS / ArcGIS Geodatabase*

4.1.2. ArcGIS

ArcGIS es un conjunto de aplicaciones para desarrollar y crear sistemas de información geográfica según los requerimientos del cliente, como aplicaciones de escritorio, servidores, aplicaciones personalizadas o aplicaciones Web. En la Figura 4.2 se presentan las opciones de desarrollo de aplicaciones utilizando ArcGIS.

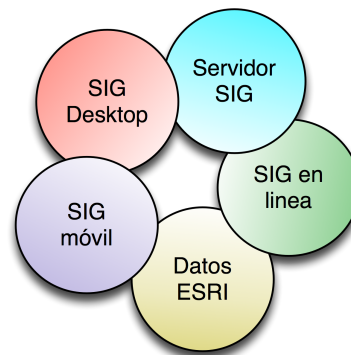


Figura 4.2: Diferentes áreas de aplicación de ArcGIS

Los productos de ArcGIS para aplicaciones de escritorio son:

ArcReader es un producto gratuito para computadoras de escritorio, de fácil uso que permite a los usuarios visualizar, explorar e imprimir mapas. Utilizando *ArcReader*, cualquiera puede ver los mapas generados utilizando productos de ArcGIS aún los mapas de alta calidad.

ArcView tiene las características de un software SIG para visualizar, administrar, crear y analizar datos espaciales. Utilizando *Arcview*, se puede entender el contexto de los datos, permitiendo observar las relaciones e identificar patrones. *ArcView* actualmente ayuda a cientos de organizaciones a tomar mejores decisiones y resolver problemas más rápidamente.

ArcEditor es sistema SIG para computadoras de escritorio para editar y gestionar datos espaciales. *ArcEditor* incluye todas las funcionalidades de *ArcView*. Además cuenta con un conjunto de herramientas para crear, editar y asegurar la calidad de los datos. *ArcEditor* soporta la edición por un solo editor así como también el trabajo colaborativo entre varios editores.

ArcInfo incluye todas las funcionalidades de *ArcEditor*, *ArcView* y *ArcReader*. Además añade herramientas para análisis espacial, manipulación de datos y herramientas de

cartografía. *ArcInfo* ofrece las herramientas suficientes para construir y administrar un SIG completo, que incluye modelos de mapas, datos, metadatos y flujos de trabajo. Tiene una interfaz intuitiva que puede adaptarse al usuario.

ArcGIS Explorer es un producto que permite de una forma muy fácil visualizar y compartir información. *ArcGIS Explorer* es gratis y de distribución libre, haciéndolo un gran medio para compartir mapas con otros.

ArcGIS Engine es un producto para desarrolladores para crear aplicaciones de escritorio SIG adaptables a las necesidades. *ArcGIS Engine* provee interfaces de programación de aplicaciones (API) para COM, .NET, Java y C++. Estas API no sólo incluyen una documentación detallada, sino también incluyen una serie de componentes visuales de alto nivel que pueden ser fácilmente reutilizados por desarrolladores para crear aplicaciones SIG.

Aplicaciones de ArcGIS para servidores:

ArcGIS Server conecta a las personas a través de aplicaciones y servicios en Web con la información geográfica. Algunas organizaciones utilizan ArcGIS Server para distribuir mapas sobre la Web a fin de mejorar flujos de trabajo internos y problemas de comunicación.

ArcGIS for AutoCAD es una herramienta gratuita, simple y de fácil uso que da a los usuarios de AutoCAD acceso a la información de SIG. Esta herramienta ofrece acceso a los servicios de *ArcGIS Server* y permite visualizar y realizar consultas geográficas dentro de los datos del ambiente de AutoCAD sin tener que transformar los dibujos CAD o convirtiendo los datos geográficos.

ArcGIS para dispositivos móviles (*Mobile GIS*) ofrece los servicios de captura, almacenamiento, actualización, manipulación, análisis y despliegue de información geográfica. Los productos que incluye *Mobile GIS* son:

ArcGIS Mobile es un SDK basado en *ArcGIS Server* para desarrollar aplicaciones personalizadas que puedan sincronizarse con el servidor

ArcPad es un software que ofrece acceder la información geográfica, mediante una base de datos, con la ayuda de dispositivos con GPS.

Los herramientas de ArcGIS para desarrollo son:

ArcGIS API para Flex permite añadir la funcionalidad de las aplicaciones enriquecidas de Internet (RIA). Con esta API puedes mostrar mapas interactivos, buscar características o atributos en SIG y mostrar los resultados.

ArcGIS API para Javascript permite utilizar *ArcGIS Server* para construir aplicaciones SIG ligeras, de alto desempeño y especialmente para navegadores Web. Con la API para *Javascript*, puedes incrustar mapas o realizar tareas como consultas espaciales sobre los datos desde una aplicación Web.

ArcGIS para Java ofrece componentes para construir aplicaciones SIG, tanto en computadoras de escritorio como en Web.

ArcGIS para .NET permite desarrollar aplicaciones en la plataforma *.Net* de *Microsoft*, como aplicaciones SIG de escritorio, para dispositivos móviles, servicios Web en la plataforma *.NET*.

4.1.3. GeoDjango

Es un *framework* para desarrollar aplicaciones Web en un lenguaje de alto nivel que es Python, éste se enfoca en un desarrollo rápido y un limpio diseño [8]. Hace uso de bibliotecas de terceros, como son GEOS (*Geometry Engine Open Source*), GDAL (*Geospatial Data Abstraction Library*) y GeoIP.

4.2. Sistemas de información geográfica en Web

En esta sección se presenta la función de los servidores de mapas y los formatos en que estos presentan los mapas. También se presentan proyectos similares al que se presenta en este trabajo de tesis.

4.2.1. Servidor de mapas

Para buscar a través de una base de datos geográfica, se utiliza un servidor de mapas, el cual permita crear imágenes de mapas geográficos, es decir, que los usuarios puedan usar el contenido directamente.

El lenguaje GML (*Geography Markup Language*) [55], es un tipo de lenguaje que describe estructuras de datos espaciales. Los datos almacenados con GML son grandes y por lo tanto lentos al transmitirse. Además, los navegadores Web no soportan GML y es necesario transformar GML a un formato visualizable en la Web.

Por otra parte, SVG toma las ventajas de imágenes vectoriales, mapas de bits y texto puro. Se puede introducir en un ambiente Web fácilmente y según SVG como formato de gráficos es factible para implementar interfaces para visualizar datos espaciales geográficos y operar en la Web [56]. En el proyecto *Geographic Information Integration and Publishing Based on GML and SVG* [56] se implementa un prototipo que permite a un usuario registrarse vía un dispositivo cliente, un servidor recibe la consulta y transmite al procesador de consultas que envía la respuesta al servidor de datos, el cual convierte los correspondientes datos espaciales en documentos GML y después se transforma en un documento SVG, que es publicado y mostrado en el navegador Web del cliente. Para el proyecto “Programación Automática de Sistemas Web a partir de Descripciones GML+x: Caso de estudio SIG” [10] la visualización de los mapas se realiza mediante SVG generado automáticamente.

4.2.2. Sistema de información turística con referencia particular a la ciudad de Huludao, China

Lichen y Chenggang [57] clarifican los conceptos, características y el método para implementar un sistema de información turística basado en un sistema Web de información geográfica. Primeramente ellos plantean un sistema Web de información geográfica como producto de la integración de la Web y las tecnologías SIG, considerándose una nueva tecnología, la cual supone crear funciones sobre SIG perfectas y diversificadas por el uso de la Internet. A diferencia de los SIG tradicionales, los sistema Web de información geográfica tienen las siguientes características:

1. Un sistema basado en Internet/Intranet que se caracteriza por la rapidez en su operación y conveniencia en mantenimiento y uso,
2. un sistema de información genuino, usado comúnmente en todo momento sin la necesidad de comprar algún software SIG,
3. el máximo nivel en promover la eficiencia en el trabajo del hardware del sistema,
4. el potencial para desarrollar un espacio en el cual se puede contribuir para establecer un amplio rango de técnicas y prácticas,
5. un bajo costo estimado en su desarrollo, uso y administración.

El caso de estudio del proyecto es un sistema de información turística para la ciudad de Huludao, compuesto por tres partes:

- Un flujo del sistema con una estructura tripartita (terminal del cliente, un servidor Web y un servidor de bases de datos),
- un diseño modular funcional con seis módulos (material básico, servicios turísticos, recursos turísticos, consultas turísticas, subsistema de herramientas y administración de los datos espaciales),
- diseño de la base de datos.

En la Figura 4.3 se muestra el flujo de trabajo en el SIG turístico para la ciudad de Huludao, China. En la Figura 4.4 se observa la estructura que da soporte al proyecto del SIG en Huludao.

Los módulos son programas en lenguaje ASP², la realización de visita a los mapas y la obtención de información, son con tecnologías *Active X* y *VisualBasic* 6.0 basándose para el desarrollo del servidor en *MapX*.

²por sus siglas en inglés de *Active Server Pages* - ASP

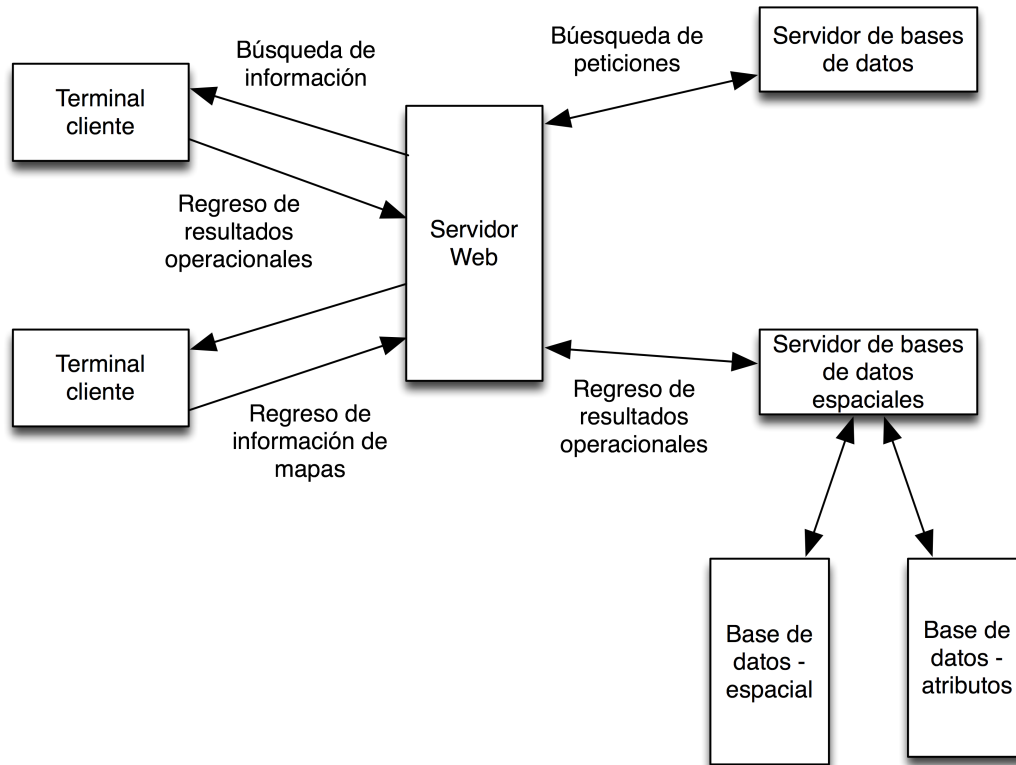


Figura 4.3: Diagrama de flujo del SIG para la ciudad de Huludao

4.2.3. Sistema de información geográfica: *DataCrossingDSS*

msCross [9] es un cliente en AJAX para Web 2.0 que parte de un sistema Web de información geográfica. Inicialmente desarrollado como interfaz para UMN *Mapserver*. Fue creado para permitir a los usuarios observar información geográfica en la Web, utilizando software libre. Otras características son: uso simple (sólo usa un archivo de *JavaScript*, no es necesaria alguna instalación), basado en AJAX, barra de herramientas fácil de manejar y soporta servicios de *Open Geospatial Consortium*.

Forma parte del proyecto *The Datacrossing DSS* [9] que es un sistema para administrar recursos sostenibles de agua, el sistema recae en un SIG distribuido para toma de decisiones a través de una interfaz amigable en la Web. La aplicación hace uso de grandes volúmenes de datos distribuidos en una infraestructura GRID. El objetivo según Manca, Cau y Bonomi, es construir una plataforma de desarrollo en la cual las ventajas computacionales y de compartición de datos que ofrecen las infraestructuras GRID promuevan trabajos que puedan cooperar con éste y formar equipos multidisciplinares.

En la Figura 4.5 se observa la distribución de SIG en Web del proyecto y las capas que ofrecen los servicios de mapeo, de visualización, de consultas, de intercambio, de procesamiento y de distribución.

Componentes importantes en este proyecto son la tecnología SRB (por sus siglas en inglés - *Storage Resource Broker*) que es un *middleware* que brinda una interfaz para

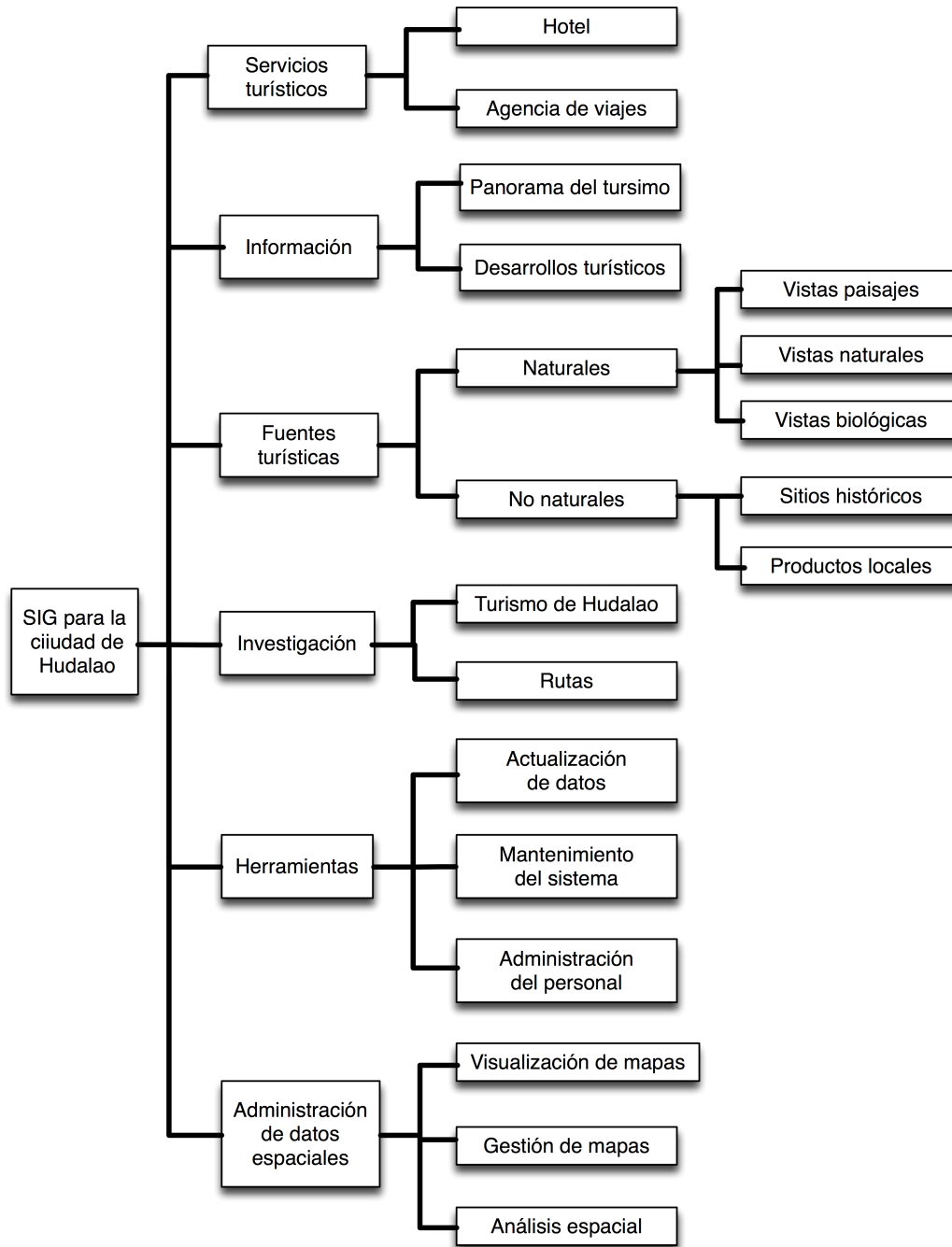


Figura 4.4: *Framework* del SIG para la ciudad de Hudalao

conectar a fuentes de datos heterogéneas sobre la red y acceder a replicas de datos. Otra parte trascendente es el administrador de servidores SRB, el componente encargado de esto es un metacatálogo (MCAT) que permite a los usuarios acceder a los recursos y datos a través de diferentes zonas. La tecnología que ocupan en la interfaz cliente está escrita en PHP y no tiene opción para usar el sistema en un ambiente para dispositivos móviles.

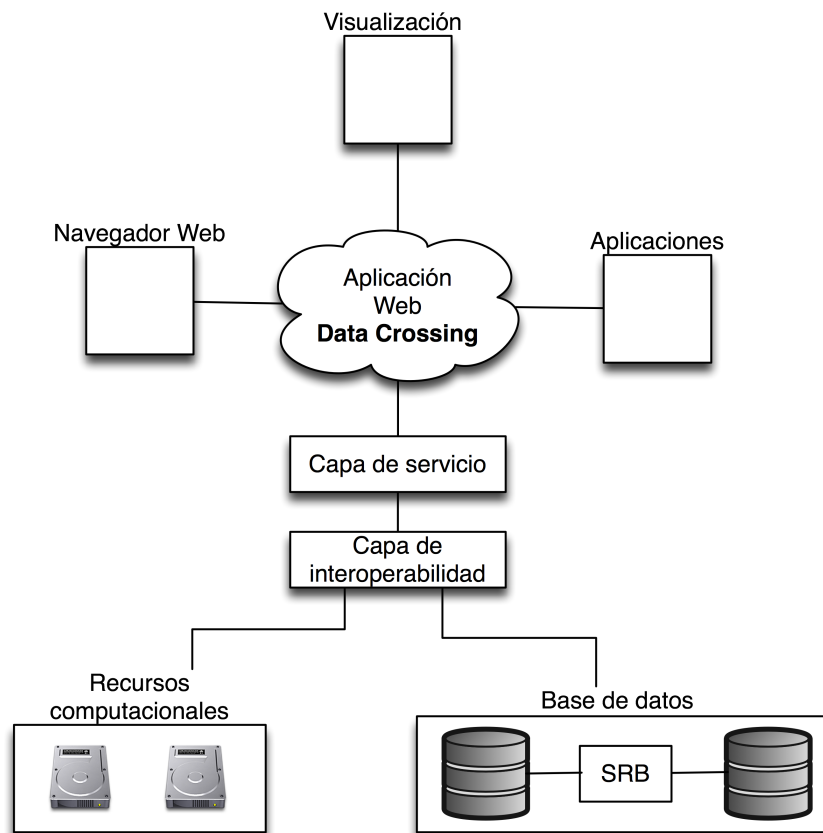


Figura 4.5: Distribución de los componentes en la Internet del sistema de *Data Crossing*

4.2.4. Programación automática de sistemas Web a partir de descripciones GML+x: caso de estudio SIG

El proyecto consiste en la construcción de una aplicación para la generación automática de código a partir de descripciones XML [23] (*eXtensible Markup Language*) y GML [55] (*Geography Markup Language*). Dichas descripciones abarcan estructura, comportamiento, interfaz y acceso de tal forma que la codificación que tengan que realizar los desarrolladores sea mínima. El caso de estudio es un Sistema de Información Geográfica, que cuenta con un modelo de datos complejos nominales y espaciales, cuya descripción se basa en el lenguaje de marcas GML, el cual puede reflejar aspectos particulares de estos sistemas. El trabajo se divide en tres aspectos fundamentales que abarcan la implementación total del proyecto. En las siguientes secciones se hace una descripción general de dichos aspectos.

Módulo de descripciones

El primer módulo consiste en el desarrollo de esquemas y descripciones que detallan todos los aspectos del sistema. Dichas descripciones se dividen en varios documentos, cada uno con una finalidad específica. De manera general las descripciones están compuestas por:

Descripción de entidades: Las entidades del sistema, son básicamente entidades existentes en la base de datos, las cuales incluyen las descripciones geográficas con características geométricas y espaciales. La base utilizada fue el estándar GML 3.1.1 que publica *Open Geospatial Consortium Inc.* que proporciona una gama de esquemas que sirven para desarrollar esquemas propietarios. De manera particular, se toma como base un esquema en específico (*features.xsd*), que a su vez hace referencia a otros esquemas del estándar. Se considera que esta base es suficiente para desarrollar las descripciones en su parte geográfica. El objetivo fue abarcar aspectos que detallen la situación de la entidad dentro de la base de datos, aspectos que serían de utilidad para el acceso a la misma.

Descripción de interfaz: Este tipo de descripción abarca propiedades y características relacionadas con la interfaz Web. En lo posible, el objetivo es detallar las características básicas de la interfaz. Se tienen en cuenta aspectos de métrica de diseño como los mostrados en el curso en línea Diseño de Interfaces Visuales. Debido a que una interfaz Web puede estar diseñada de muchas maneras, es un tanto difícil realizar descripciones de tales características. Por tal motivo, se propone un diseño de interfaz específico que hasta cierto punto sea parametrizable. De esta manera, la interfaz puede ser descrita de manera más general. Para describir la interfaz es necesario dividir el sistema en módulos. A cada módulo se le asocia una interfaz principal. Como se mencionó anteriormente la interfaz está predefinida en cuanto a su estructura principal. Consta de un encabezado de página, un menú, un submenú y la parte de contenidos.

Descripción de menús: A cada uno de los módulos que se han identificado en el sistema se les asocia un menú, ésto con la finalidad de hacer una mejor organización de las funciones que proveerá el sistema. Cada menú de los módulos está descrito en un archivo de descripción donde se especifica cada componente del menú y el enlace asociado. Dentro de esta descripción también se especifican características de presentación, tales como color, fuente, tamaño de fuente, etc. Generalmente cada opción del menú tiene un enlace hacia un submenú por lo que también se realizan descripciones de submenús para cada opción de cada módulo identificado.

Descripción de comportamiento: Dentro de estas descripciones se abarca el detalle del comportamiento del sistema. Aquí se describe la forma en la que se ha de navegar en el sistema. La descripción de comportamiento está dividida en módulos de la misma forma en que se han identificado los módulos en el sistema. Cada módulo se divide en entidades y cada entidad en modos de comportamiento (especificados también en las opciones de los submenús)

Descripción de mapas: Debido a que el caso de estudio es un Sistema de Información Geográfica, es fundamental la generación de mapas. Se cuenta con descripciones que detallan la manera en que han de desplegarse dichos mapas en las páginas Web. Es importante mencionar que para dar interactividad al usuario se cuenta con funciones predefinidas de *JavaScript* que proporcionan principalmente funciones de acercamiento, alejamiento y selección de entidades geométricas.

Módulo generador de código

El siguiente módulo consiste en la implementación del generador de código. Esta actividad es la más extensa y su complejidad radica en la asociación del modelo plasmado en las descripciones a código interpretable por un navegador. El generador de código está desarrollado en Java en conjunto con JDOM (*Java Document Object Model*) y *xTags* (*Tag Libraries*) para el manejo de documentos y transformaciones a código HTML (*HyperText Markup Language*) o SVG (*Scalable Vector Graphics*). El generador de código lee las descripciones y las convierte en código que pueda ser compilado y ejecutado. El generador de código se auxilia de plantillas que consisten de métodos prefabricados que realizan varias tareas y de hojas de estilo para las transformaciones a HTML o SVG. La razón por la cual se ha elegido trabajar con plantillas es debido al hecho de que describir un método resulta complicado, puesto que un método puede realizar una gran variedad de funciones y de alguna manera es difícil que el generador interprete las descripciones que son de por sí complicadas. De esta manera, los métodos contenidos en plantillas son parametrizables, es decir, reciben parámetros que hasta cierto punto determinarán la función que realizará (*e.g.* indicar el tipo de dato que regresará). Ahora bien, las plantillas de hojas de estilo consisten en código XSLT (*eXtensible Stylesheet Language Transformations*) que puede programarse para que las descripciones XML se transformen en código HTML embebido en las páginas Web. Básicamente se tendrá código XSLT que involucre formularios y elementos de dichos formularios, menús y submenús. De esta manera el generador lee las descripciones, utiliza las plantillas XSLT previamente definidas y se realiza la transformación para generar código HTML embebido dentro de una JSP.

Módulo de aplicación final

Este módulo representa la aplicación final en donde se obtiene el código generado incluyendo clases, código *Javascript*, código SVG y páginas JSP organizado en directorios de forma tal que sea posible agregar la aplicación a un contenedor de aplicaciones Web y que pueda ejecutarse desde un navegador. De manera general, el sistema proveerá las siguientes funcionalidades:

- Autenticación de usuario,
- subsistema de consulta y captura de datos nominales y espaciales. Éstos serán para la capa de hidrología de la base de datos geográfica del estado de Colima
- visualización de mapas en la capa de hidrología,
- Interacción del usuario con el sistema para la captura y funciones de visualización (selección de entidades geográficas).

Resumen del capítulo

En la primera sección de este capítulo se presentaron API para el desarrollo de sistemas de información geográfica, como: *Mapaccel*, *ArcGIS* y *GeoDjango*. *Mapaccel* es un *framework* que permite crear SIG complejos para consultar o editar datos espaciales, Las desventajas que tiene son su licencia propietaria y su fuerte interdependencia entre las

aplicaciones de *Mapacel*, lo que hace un *framework* un poco versátil para los desarrolladores. Por otra parte, está *ArcGIS*, que es un conjunto de herramientas para desarrollar SIG tanto en ambiente Web como en PC. Aunque algunos de los productos son gratuitos, de igual forma que *Mapacel*, existe una interdependencia entre estos productos, además que cuando uno desea crear SIG más complejos se necesita licencia propietaria. Y tiene como inconveniente que los productos sólo se disponen para PC con sistema operativo *Windows*. Una parte interesante de *ArcGIS* es lo desarrollado para dispositivos móviles, que contempla aplicaciones Web y aplicaciones SIG para instalar en los dispositivos. Y se puede programar con lenguaje Java o sobre la plataforma de *Microsoft .NET*. Sobre *GeoDjango* se tiene que es un *framework* para desarrollar SIG que con el soporte del lenguaje de programación de alto nivel *python*, que se enfoca en un desarrollo rápido y un diseño limpio.

En la segunda sección se presentan sistemas de información geográfica en Web. Primero se describen los formatos en los cuales se representan los mapas desde un servidor y el lenguaje de marcas GML que describe estructuras de datos espaciales. Después, se describieron tres SIG implementados para un ambiente Web. El primero es un SIG para turismo para la ciudad de Huludao, en China que desarrolla un SIG en Web con las tecnologías de: *ASP*, *Active X* y *VisualBasic*. El segundo proyecto es un SIG llamado *DataCrossingDSS* en el cual se tiene un módulo para ofrecer la visualización de mapas en navegadores Web con las tecnologías de AJAX y sobre el paradigma Web 2.0. El tercer proyecto se realizó para generar SIG de forma automática a partir de descripciones en GML, pero éste sólo lo adapta para navegadores Web en PC de escritorio.

En el próximo capítulo se presentan el diseño y el desarrollo de la biblioteca de *tags* JSP como componentes reutilizables, con la ventaja de ser una herramienta fácil de comprender para los desarrolladores de SIG en Web. También se presenta la estructura de la biblioteca y se describen cada uno de los *tags* creados. Se describe el diseño de dos sistemas de información geográfica para validar la biblioteca, uno para el desarrollo sustentable en hidrología y otro para el apoyo a la toma de decisiones en la infraestructura educativa.

Capítulo 5

Diseño y desarrollo

En este capítulo se presenta nuestro diseño y el desarrollo del proyecto de tesis. Tal como lo planificamos, la API está compuesta de clases de Java, de código JavaScript y de una biblioteca de *tags* para JSP. Estos elementos permiten a desarrolladores de aplicaciones Web crear SIG dentro del paradigma Web 2.0, ya que la API proporciona métodos para:

- Realizar conexiones a bases de datos con el SGBD,
- realizar consultas geográficas,
- visualizar objetos geográficos,
- ofrecer métodos para la transferencia asíncrona de mensajes entre el servidor y el cliente con la incorporación del objeto *XMLHttpRequest* (AJAX).

La API de SIG en Web para dispositivos móviles se divide en módulos de consulta y visualización de los objetos geográficos. Para un adecuado establecimiento del marco de trabajo se realiza el diseño arquitectónico del sistema. Se realiza el modelado UML estructural para definir los componentes que conforman al sistema, y el modelado UML de comportamiento para conocer el flujo de actividades, mensajes y los roles de los elementos dentro del sistema.

5.1. Metodología de diseño y desarrollo

El diseño y desarrollo está orientado a la reutilización de componentes [58], que pueden ser requeridos e incorporados en otros sistemas de información geográfica. Se eligió este enfoque de desarrollo debido a que la intención primaria es aportar una herramienta para la creación de sistemas Web de información geográfica. En este trabajo de tesis se realizan las validaciones en el navegador Web Safari en su versión para los dispositivos móviles *iPodTouch/iPhone* de Apple. Este enfoque también se basa en componentes existentes [59].

En la Figura 5.1 se muestran las etapas del enfoque de desarrollo orientado a la reutilización, las más destacadas en el proceso orientado a la reutilización son [58]:

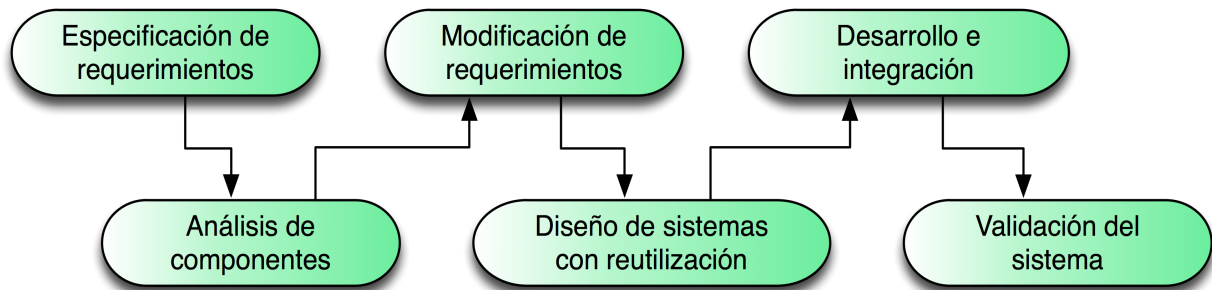


Figura 5.1: Etapas para el diseño y desarrollo orientado a la reutilización

Especificación de requerimientos - La etapa de especificación de requerimientos cubre el periodo para coleccionar los servicios y restricciones de los sistemas, para que posteriormente, éstos sean analizados y verificados

Análisis de componentes - En esta etapa se propone definir los requerimientos de los componentes para determinar su funcionalidad específica.

Modificación de requerimientos - Durante el periodo de diseño y desarrollo, los requerimientos pueden modificarse debido a cambios funcionales. Los componentes generados de esos cambios son los que están ahora disponibles a reutilizar. Si las modificaciones no son posibles la etapa de análisis de componentes se lleva a cabo nuevamente.

Diseño de sistemas con reutilización - En esta fase se diseña un marco para el nuevo sistema. Para el diseño se contemplan los componentes reutilizables. Además de organizar el sistema para que se ajuste a ellos.

Desarrollo e integración - Los componentes y subsistemas se integran para crear un sistema. En esta metodología la integración de sistemas es parte del proceso de desarrollo, más que una actividad separada.

Validación del sistema - Esta etapa incluye la comprobación y análisis de los componentes para asegurar que cumplan con el objetivo del trabajo, el cual es ofrecer una herramienta para visualizar objetos geográficos en sistemas Web de información geográfica en dispositivos móviles.

La etapa de especificación de requerimientos se especifica con el apoyo del modelado UML, véase sección 5.7, y la etapa de validación del sistema, se muestra en el capítulo 6. En las siguientes subsecciones se describen las etapas del proceso para el desarrollo SIG en Web en los dispositivos móviles.

5.2. Análisis de componentes en un SIG

Este trabajo propone la generación dinámica de los datos espaciales en gráficos Canvas y SVG para su visualización en navegadores Web. Se propone una biblioteca de *tags* JSP

propia para generar los gráficos y consultar al sistema de base de datos espacial. Con el fin de facilitar a los desarrolladores la inclusión de componentes para visualizar objetos geográficos en un SIG en la Web. Para fundamentar esta idea, en los siguientes párrafos se presenta la estructura de la biblioteca y sus ventajas.

Biblioteca de *tags* JSP

Una biblioteca de *tags* abstrae la funcionalidad que manejan las JSP¹ para definir un “lenguaje”² que permita de forma más natural crear aplicaciones Web. Dentro las ventajas que da el uso de bibliotecas de *tags*, podemos poner en evidencia que:

- No es necesario que el desarrollador conozca lenguaje Java,
- es fácil de añadir a los *frameworks* para desarrollar aplicaciones Web,
- separación de la “lógica de negocio”³ y la presentación [29], es decir, separar los datos del sistema, de las páginas Web a través de cierta lógica para crear comunicación entre el sitio Web y el sistema de base de datos.

La descripción técnica en detalle se especifica en el descriptor de la biblioteca de los *tags* (TLD, *Tag Library Descriptor*, en inglés) el cual se explica en la subsección 5.6. Para la biblioteca desarrollada en esta tesis el descriptor es llamado `giswebtag.tld` y para utilizarlo debe colocarse en la carpeta `WEB-INF` de la aplicación Web y darse de alta en el descriptor de la aplicación Web (`web.xml`) que se encuentra en la misma carpeta añadiendo las siguientes líneas:

```
(1)<taglib>
(2)  <taglib-uri>WEB-INF/tags</taglib-uri>
(3)  <taglib-location>/WEB-INF/giswebtag.tld</taglib-location>
(4)</taglib>
```

En las líneas (1) y (4) está la etiqueta para indicar la biblioteca de *tags*, en la línea (2) se indica el identificador (URI⁴) de la biblioteca y en la línea (3) se indica la localización del descriptor de la biblioteca.

5.3. Generación dinámica de objetos geográficos

Con el fin de crear una aplicación Web ligera⁵ se desarrolla una API, para la generación dinámica de objetos geográficos en formatos gráficos SVG y Canvas. Los objetos geográficos se crean en el dispositivo móvil, es decir, en el navegador Web y nunca hay un

¹Java Server Pages

²También podría manejarse como sublenguaje

³Traducción de la frase en inglés: *business logic*

⁴*Uniform Resource Identifier*

⁵Que tenga la característica de que su contenido sea el mínimo necesario y de rápido tiempo de descarga

almacenamiento del lado del servidor. En los siguientes párrafos se describe la generación de los objetos geográficos en los formatos gráficos Canvas y SVG.

Generación dinámica de gráficos Canvas - El objeto geográfico se presenta mediante un gráfico Canvas⁶ en los dispositivos móviles de Apple. La definición del elemento Canvas se define con código *JavaScript*. Por lo tanto, se crea un conjunto de *tags* para generar código *JavaScript* dinámicamente, este código contiene los objetos geográficos que resultan de la consulta del usuario. Estos objetos geográficos pueden ser de cualquiera de las capas del SIG que se tengan en la base de datos geográfica (corrientes de agua, carreteras, curvas de nivel, edificios, lagos, límites, entre otros).

Generación dinámica de gráficos SVG - También se generan objetos geográficos con SVG debido a que este formato gráfico permite visualizar imágenes vectoriales sobre las cuales se puede realizar acercamientos, sin afectar de forma negativa la resolución del gráfico, mientras que con el formato Canvas se trabaja con píxeles y en los acercamientos se pierde resolución. En la Tabla 5.1 se enlistan los navegadores Web que soportan gráficos SVG y Canvas.

Navegador Web
Internet Explorer
Mozilla Firefox
Safari
Safari Mobile
Opera

Tabla 5.1: Navegadores Web que soportan los formatos gráficos SVG y Canvas.

Requerimientos en los dispositivos móviles

Como se ha mencionado, los dispositivos móviles sobre los cuales se diseñó el sistema son el *iPodTouch* y el *iPhone* de Apple. Debido a que soportan a través de su navegador Web *Safari* en su versión para móviles las tecnologías para el paradigma Web 2.0.

Entonces se requiere que los dispositivos móviles tengan un navegador que soporte gráficos SVG o Canvas y de preferencia que tenga funciones de acercamiento y alejamiento con pantalla sensible al tacto. El requerimiento esencial para la API que desarrollé no es tanto la capacidad de memoria, o los servicios de telefonía, si no, que el navegador Web que tenga el dispositivo y la velocidad de transmisión de los datos. Los navegadores Web en su versión para dispositivos móviles que soportan SVG o Canvas, son: *Opera Mobile*, *Skyfire* y *Safari Mobile*⁷.

⁶Canvas Tag HTML 5

⁷Sitios Web de los navegadores Web, <http://www.opera.com/mobile/>, <http://www.skyfire.com/>, y <http://www.apple.com/iphone/features/safari.html>

5.4. Modificación de requerimientos

Uno de los requerimientos de los SIG para la Web 2.0, es la asincronía entre el servidor y el navegador Web del dispositivo móvil mediante el objeto *XMLHttpRequest* de AJAX. El requerimiento anterior es uno de los objetivos del paradigma Web 2.0, ya que mediante la asincronía se pueden crear aplicaciones Web que ofrezcan una rápida velocidad de respuesta al usuario, tal como si fuera una aplicación de escritorio.

Los módulos desarrollados para los SIG de validación, utilizan el *framework Prototype* [60] para manejar la asincronía; para los efectos y visualización se ocupa el *framework Scriptaculous* [61], ambos *frameworks* están en código *JavaScript*. Además se hace uso de la biblioteca de *Tags* JSP estándar (JSTL [62]) como complemento a los *tags* que se crearon para ser reutilizables, ya que con JSTL ya se pueden crear instancias de conexiones a bases de datos y realizar consultas mediante lenguaje SQL.

5.4.1. Arquitectura de la biblioteca

La biblioteca de *tags* para JSP a desarrollar deber ser adaptable a *frameworks* para aplicaciones Web enfocados al lenguaje de programación Java. Algunos *frameworks* son *Struts* [63], *Java Server Faces* [64], *Spring* [65] y *Tapestry* [66] entre otros.

En la Figura 5.2 se presenta el esquema general de un SIG en Web, y a continuación se describen los pasos y el camino que toma cada petición que se realiza desde el cliente hasta el servidor.

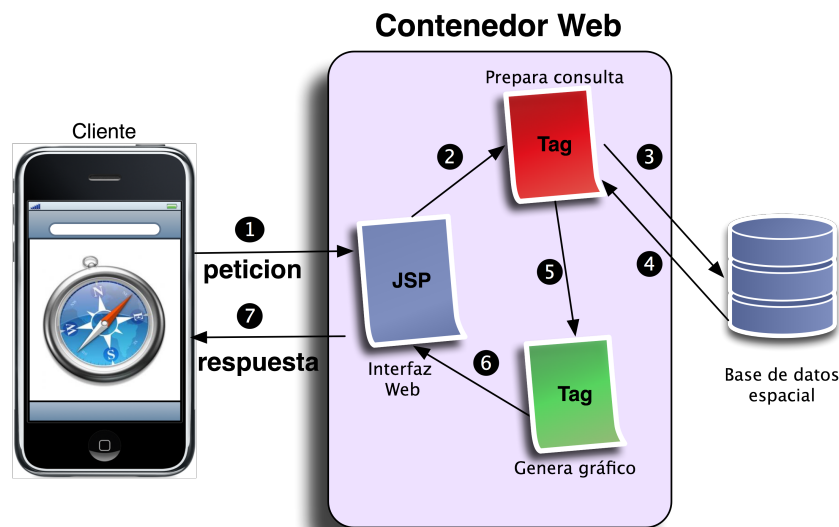


Figura 5.2: Esquema general propuesto para la creación de gráficos en navegadores Web

1. A través del navegador Web se realiza una petición HTTP hacia el contener Web del servidor,

2. la JSP toma los parámetros para las consultas y se prepara la consulta deseada utilizando el *tag* que contiene los tipos de consultas espaciales SQL,
3. dentro del *tag* se crea una instancia para la conexión a la base de datos espacial y ejecuta la consulta,
4. la instancia de conexión regresa un objeto con la respuesta de la consulta,
5. con código *JavaScript* se recuperan los valores resultantes y se envían al *tag* que forma los gráficos,
6. el código *JavaScript* es incrustado y evaluado por la JSP,
7. la JSP incrusta el código que forma los gráficos de los objetos geográficos al documento HTML para ser visualizados en el navegador Web del cliente.

El *tag* para la conexión a bases de datos permite comunicar al sistema Web con la base de datos espacio- temporal; el *tag* de consultas de toponimia⁸ y de objetos geográficos permite consultar los atributos de los objetos geográficos, con la ventaja de que los atributos de entrada para consultar es simple y no es necesario conocer el lenguaje SQL; el *tag* de los enlaces permite hacer uso de otras API (como *Google Chart* [67]); también están los *tags* que generan dinámicamente los objetos geográficos en formato Canvas y SVG. En la Figura 5.3 se muestra el árbol de la biblioteca de *tags* creados.

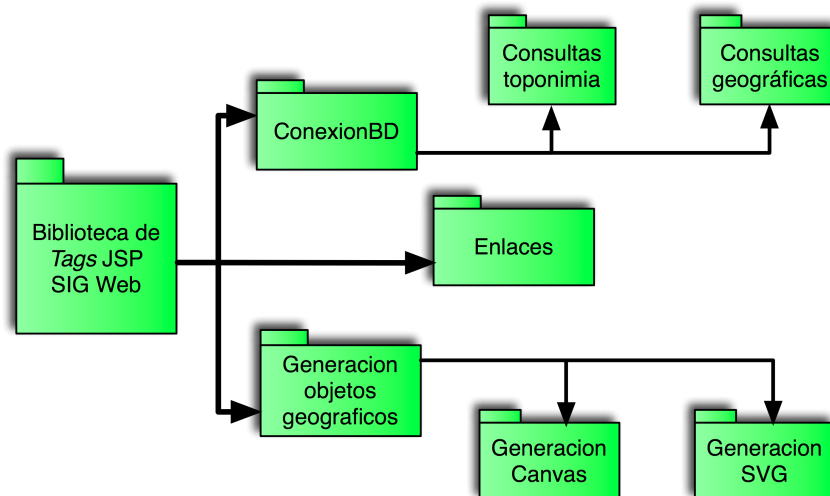


Figura 5.3: Estructura de la biblioteca de *tags*

⁸Se refiere al nombre oficial de un espacio geográfico.

5.4.2. Herramientas adicionales

A continuación se describen las aplicaciones que se utilizan para el funcionamiento de la API:

PostgreSQL / PostGIS [47] [48] - se utiliza para soportar bases de datos espaciales o geográficas en el sistema gestor de bases de datos *PostgreSQL*. *PostGIS* contiene funciones específicas para realizar consultas geográficas entre objetos, como la área o perímetro y la distancia entre objetos.

Prototype / Scriptaculous [60] [61] - proporcionan soporte para el comportamiento asíncrono de los documentos HTML basado en AJAX. Además permiten el control de los eventos en el documento y la inclusión de efectos en las páginas Web.

Biblioteca de *tags* JSP estándar (JSTL) [62] - da soporte para realizar operaciones básicas en JSP a través de *tags*. El uso de *tags* facilita el desarrollo de JSP porque permite utilizar condiciones, realizar conexiones a bases de datos y utilizar lenguaje SQL para consultas.

Las aplicaciones descritas anteriormente son necesarias y se acoplan a los requerimientos. *PostgreSQL / PostGIS* se utilizan debido a que la base de datos espacio temporal se encuentra en el SGBD *PostgreSQL*, que junto con *PostGIS* como aplicación para soporte de bases de datos espacial permite la utilización de funciones para consultar objetos geográficos y atributos de ellos; *Prototype / Scriptaculous* son API en *Javascript* que facilitan la incorporación de AJAX al sistema Web, es decir, permiten asincronía en las peticiones entre el servidor Web y el cliente; y la biblioteca de *tags* JSP estándar que encapsula funciones comunes en sistemas Web dinámicas, como iteraciones, condiciones y realización de consultas por medio de lenguaje SQL a bases de datos.

5.5. Diseño de sistemas con reutilización de componentes

Para la validación de los componentes creados en el presente trabajo, se diseñaron dos sistemas Web. El primero es un complemento a un sistema Web de información geográfica para el desarrollo sustentable en el campo de la hidrología realizado por Hernández [10]. El segundo surge como parte de una propuesta de proyecto para la Secretaría de Educación Pública (SEP) por parte del Dr. Sergio Chapa Vergara. En las siguientes subsecciones se presentan los diseños de los SIG.

5.5.1. Sistema Web de información geográfica para el desarrollo sustentable en hidrología

Este proyecto resuelve crear un SIG en Web 2.0 para visualizar y administrar objetos geográficos en el campo de la hidrología y apoyar el desarrollo sustentable.

Descripción

El sistema almacena dos tipos de datos: datos geográficos, dados por un conjunto de mapas y datos climáticos como la temperatura, presión y humedad entre otros, estos últimos se recuperan usando sensores. El SIG en Web actualiza la base de datos geográfica para su consecuente aparición de ellos en las próximas consultas.

El sistema combina bases de datos espacio temporal y bases de datos geográficas para lograr el objetivo del proyecto, que es mejorar la regulación de los recursos hidrológicos en México. En específico, se realizó con datos espaciales del estado de Colima proporcionados por INEGI⁹.

Debido a que los navegadores Web de los dispositivos móviles de Apple (*iPhone/iPodTouch*) soportan las últimas tecnologías Web, el SIG está diseñado especialmente para estos. El SIG genera dinámicamente los objetos geográficos en formato Canvas y SVG.

Además, el SIG muestra una gráfica de los valores medidos por los sensores a través del tiempo. Los nodos de medición son localizados estratégicamente en ríos, lagos o mares. Estos nodos envían datos que son almacenados en la base de datos espacio temporal.

Para la visualización de los objetos geográficos en el navegador Web se requieren los *tags* propios para la generación de gráficos y la biblioteca de *tags* JSTL para realizar la conexión a la base de datos espacial. La comunicación entre el servidor y el cliente debe ser asíncrona, por lo que se incluye el *framework Prototype*. Además del uso de la API *Google Chart*.

Parte del sistema son las vistas o interfaces del usuario, las cuales se engloban en el componente de “visualización y administración de objetos geográficos” con el estereotipo «*application*». Otro componente es la biblioteca de “*tags* JSTL” que realiza la conexión al componente “base de datos geográfica-espacial de Colima” con estereotipo «*database*». Se crea el componente para la “transmisión asíncrona” que hereda su funcionalidad del *framework Prototype* el cual es un componente de estereotipo «*framework*». El componente de “*tags* JSP propios” se conecta con el de la “transmisión asíncrona” para la generación de los gráficos. El componente “API Google Chart” se utiliza para mostrar gráficas de las mediciones en los sensores almacenadas en la base de datos a través del tiempo. La Figura 5.4 muestra los componentes para la creación del sistema.

⁹Instituto Nacional de Estadística y Geografía, www.inegi.gob.mx

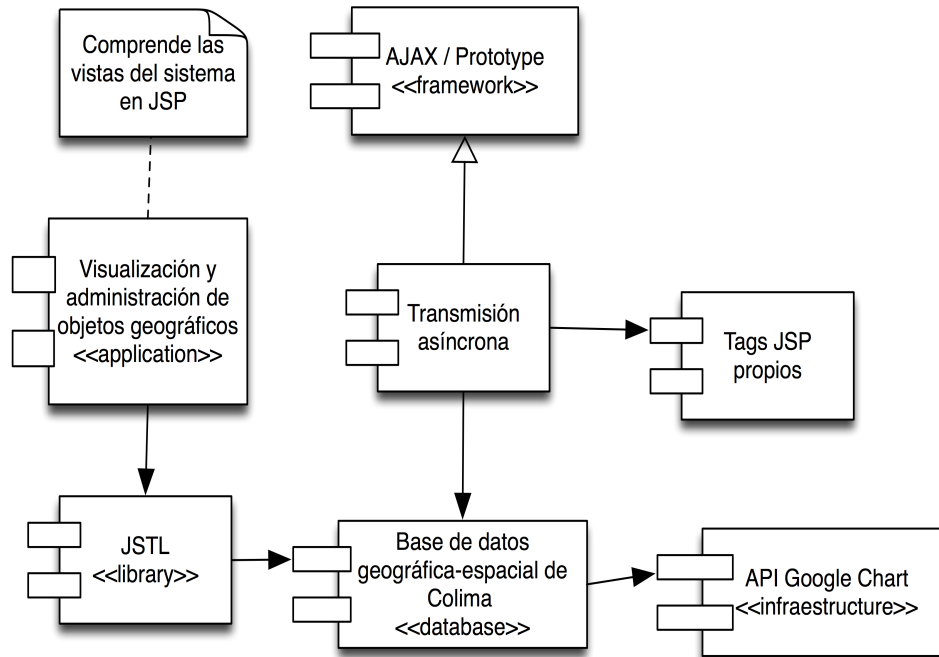


Figura 5.4: Diagrama de componentes del SIG en Web para el desarrollo sustentable en hidrología

5.5.2. Sistema Web de información geográfica para la toma de decisiones en la infraestructura educativa

Descripción

El sistema da soporte a la toma de decisiones para la planeación de infraestructura educativa, ya que permite visualizar lugares geográficos de las escuelas o institutos de educación. Así mismo permite mostrar objetos geográficos de la zona cercana a la escuela. Estos objetos geográficos se categorizan en capas, como son: hidrología, vías de comunicación, topografía, zonas rurales y urbanas entre otras.

Principalmente, se almacenan datos espaciales de los objetos geográficos y de las escuelas en una base de datos espacial. Y por otra parte, se almacenan los atributos toponímicos correspondientes a las escuelas para conocer el nombre oficial de la escuela, dependencia, estado de la construcción; y algunos datos como: nombre del director, número de estudiantes, turno y fecha de apertura.

Al poder visualizar las escuelas y los objetos geográficos de la misma zona geográfica se da soporte para la planeación de infraestructura y la toma de decisiones para la construcción de centros educativos.

El conjunto de vistas del sistema Web se representan por el componente “visualización y administración de objetos geográficos” que hace uso del componente “*Scriptaculous*”, con estereotipo «framework», el cual ofrece mayor interactividad entre el navegador Web y el usuario. El componente para la “transmisión asíncrona” hereda del *framework Prototype*,

esta transmisión involucra los componentes: “JSTL” y “*tags* JSP propios” que son el conjunto de *tags* para generar el gráfico en el navegador Web; el componente “Base de datos espacial” almacena los datos espaciales, y tiene estereotipo «*database*». También es necesario una base de datos para almacenar las imágenes de los centros educativos y sus atributos, por lo que surge el componente *base de datos de fotos y atributos*. La Figura 5.5 muestra los componentes para la creación del sistema.

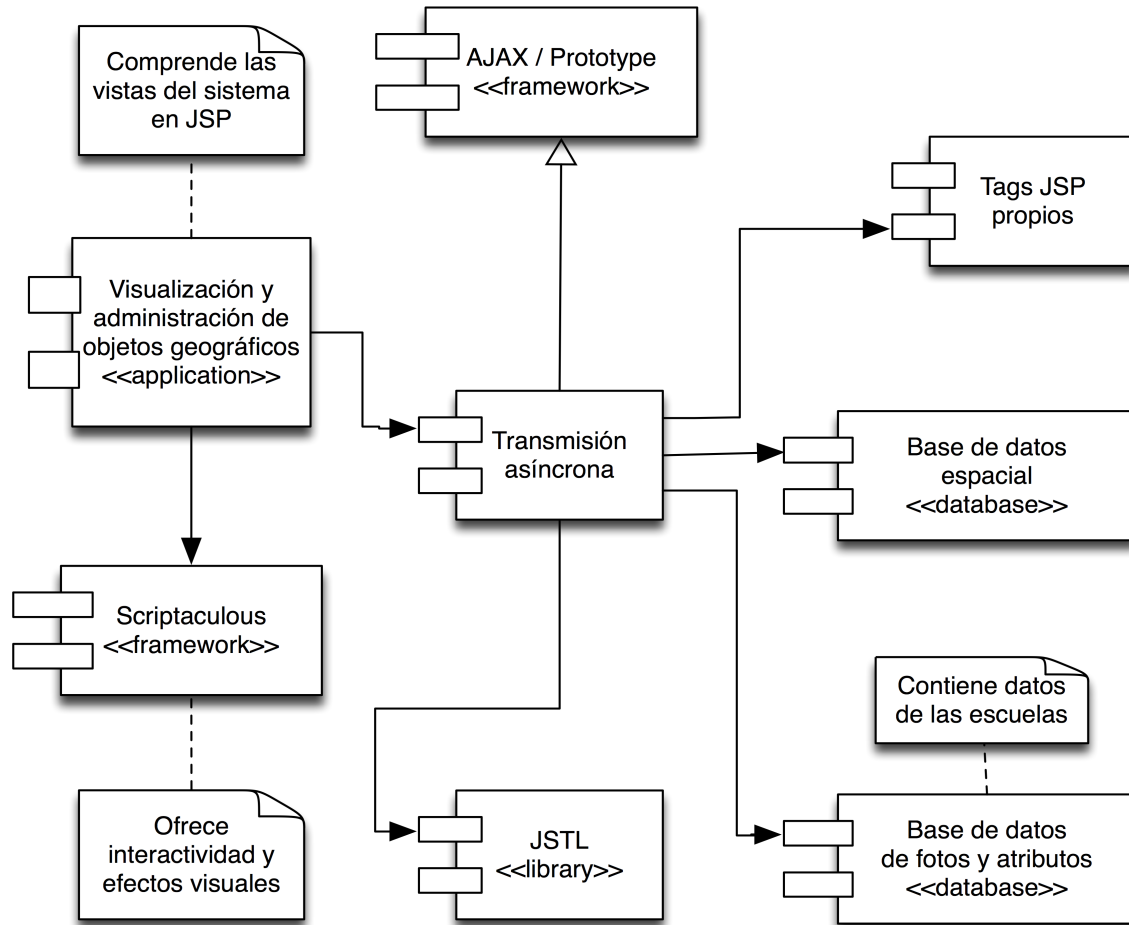


Figura 5.5: Diagrama de componentes del SIG en Web para la toma de decisiones en la infraestructura educativa

5.6. Desarrollo e integración

La metodología para desarrollar las bibliotecas de *tags* fue enfocada a la versión JSP 2.0, en la que se crea una carpeta llamada *tags*, que contiene los archivos **.tag* dentro de la carpeta WEB-INF de la aplicación Web. Para crear la librería de *tags* JSP para desarrollar el SIG en Web y para compartirla sólo es necesario definir el archivo **.tld* (*Tag Library Descriptor*) el cual contiene una estructura XML con la descripción de cada *tag* y los

parámetros que recibe.

La descripción de cada *tag* se da de la siguiente manera:

Nombre se define dentro de la etiqueta `< name >`.

Clase se define para manejar el *tag* en el caso que sea necesario.

Contenido se describe el contenido del *tag*. El contenido del *tag* puede ser *scriptless*, lo que significa que no se utiliza código Java (*scriptlets*). El caso en que el *tag* no utiliza su contenido, se especifica con *empty*.

Atributos La descripción de cada atributo es usando la etiqueta `< attribute >` y dentro de esta etiqueta los siguientes *tags*:

Nombre de atributo el nombre del atributo se define con `< name >`.

Tipo de atributo se define cuando un atributo es necesario u opcional, *required* cuando es necesario y *false* cuando es opcional. Se utiliza la etiqueta `< required >`.

Evaluación cuando es necesario evaluar expresiones de otros JSP como valores del atributo, se utiliza la etiqueta `< rtexprvalue >` con valor *true* y cuando los valores son mezclados se coloca *false*.

Definición del descriptor de la biblioteca de *tags* - A continuación se presenta un ejemplo para definir la estructura del descriptor de la biblioteca de *tags*:

```
<taglib ...>
  ...
  <tag>
    <name>nombre_clase</name>
    <tag-class>jsp.tags.nombre_clase</tag-class>
    <body-content>empty</body-content>
    <attribute>
      <name>nombre_atributo</name>
      <required>>true</required>
      <rtexprvalue>>false</rtexprvalue>
    </attribute>
    <attribute>
      <name>nombre_atributo1</name>
      <required>>false</required>
      <rtexprvalue>>false</rtexprvalue>
    </attribute>
  </tag>
  ...
</taglib>
```

5.6.1. *Tags* JSP creados para su reutilización en SIG

Para explicar como se pueden reutilizar la biblioteca de *tag* JSP, se describe cada *tag* mediante su correspondiente estructura XML en el archivo descriptor de la biblioteca de *tags* (TLD).

Tag - geoqueries

Este *tag* fue desarrollado para obtener las consultas espaciales que se aplicarán a la base de datos, con la ventaja que las consultas espaciales obtenidas ya están formadas con los parámetros que recibe el *tag* y listas para ser ejecutadas a la base de datos. Para el usuario o desarrollador no es necesario saber crear consultas espaciales, sólo es necesario definir los parámetros de entrada al *tag*.

Descriptor de la biblioteca

```
(1)<tag>
(2)  <description>
(3)    Ofrece sentencias SQL para ejecutarse en la base de datos.
(4)  </description>
(5)  <name>geoqueries</name>
(6)  <body-content>scriptless</body-content>
(7)  <attribute>
(8)    <description>
(9)      Define la consulta espacial.
(10)    </description>
(11)    <name>type</name>
(12)    <required>>true</required>
(13)    <rtexprvalue>>false</rtexprvalue>
(14)  </attribute>
(15)  <attribute>
(16)    <description>
(17)      Determina la distancia entre objetos geograficos.
(18)    </description>
(19)    <name>distance</name>
(20)    <required>>false</required>
(21)    <rtexprvalue>>false</rtexprvalue>
(22)  </attribute>
(23)  <attribute>
(24)    <description>
(25)      Define el nombre de la tabla base de datos.
(26)    </description>
(27)    <name>table</name>
(28)    <required>>false</required>
(29)    <rtexprvalue>>false</rtexprvalue>
(30)  </attribute>
```

```
(31)     <attribute>
(32)         <description>
(33)             Define el nombre de una 2da tabla en la base de datos.
(34)         </description>
(35)         <name>table1</name>
(36)         <required>>false</required>
(37)         <rtexprvalue>>false</rtexprvalue>
(38)     </attribute>
(39)     <attribute>
(40)         <description>
(41)             Define el nombre de una 3a tabla en la base de datos.
(42)         </description>
(43)         <name>table2</name>
(44)         <required>>false</required>
(45)         <rtexprvalue>>false</rtexprvalue>
(46)     </attribute>
(47)     <attribute>
(48)         <description>
(49)             Define el identificador del objeto geografico.
(50)         </description>
(51)         <name>fid</name>
(52)         <required>>true</required>
(53)         <rtexprvalue>>false</rtexprvalue>
(54)     </attribute>
(55) </tag>
```

Descripción

Entre las líneas (1) y (55) se declaran las características del *tag* y la descripción de sus atributos. En la línea (3) se encuentra la descripción general del *tag*, que se declara dentro de la etiqueta `<description>` (líneas (2) y (4)). En la línea (5) se define el nombre del *tag* y en la línea (6) se define que el contenido del *tag* no tiene código Java, colocando la palabra *scriptless*. De la línea (7) a la (14) se define el atributo llamado *type* en la línea (11) con su descripción en las líneas (8), (9) y (10), este atributo es requerido de forma obligatoria lo que se indica en la línea (12) y en la línea (13) se declara que no se evaluarán expresiones de otras JSP como valor de este atributo.

De la línea (15) a la (22) se declara el atributo llamado *distance*, línea (19), refiriéndose a una distancia entre objetos geográficos, este atributo se ocupa en varias consultas pero no es obligatorio, así lo indica la línea (20). La descripción está de la línea (16) a la (18). Y no evalúa expresiones de otras JSP y se indica en la línea (21).

Los atributos llamados *table* (de la línea (23) a la (30)), *table1* (de la línea 31 a la (38)) y *table2* (de la línea 39 a la (46)) definen nombres de las tablas. El atributo llamado *fid* (de la línea (47) a la (54)) define el identificador del objeto geográfico. La forma de leer la descripción de estos últimos atributos del *tag* es parecida a la de los primeros. Cabe mencionar que el nombre de la base de datos se declara en los *tags* de la biblioteca JSTL.

Tag- geostd

Existen consultas generales determinadas por el OGC (*Open Geospatial Consortium*). Estas se encuentran en este *tag*.

Descriptor de la biblioteca

```
(1)<tag>
(2)  <description>
(3)    Ofrece sentencias SQL no espaciales.
(4)  </description>
(5)  <name>geostd</name>
(6)  <body-content>scriptless</body-content>
(7)  <attribute>
(8)    <description>
(9)      Determina la distancia entre objetos geograficos.
(10)    </description>
(11)    <name>distance</name>
(12)    <required>>false</required>
(13)    <rtexprvalue>>false</rtexprvalue>
(14)  </attribute>
(15)  <attribute>
(16)    <description>
(17)      Define el identificador del objeto geografico.
(18)    </description>
(19)    <name>gid</name>
(20)    <required>>false</required>
(21)    <rtexprvalue>>false</rtexprvalue>
(22)  </attribute>
(23)  <attribute>
(24)    <description>
(25)      Define el nombre de la tabla.
(26)    </description>
(27)    <name>table</name>
(28)    <required>>false</required>
(29)    <rtexprvalue>>false</rtexprvalue>
(30)  </attribute>
(31)  <attribute>
(32)    <description>
(33)      Define el tipo de objeto geografico.
(34)    </description>
(35)    <name>geotype</name>
(36)    <required>>false</required>
(37)    <rtexprvalue>>false</rtexprvalue>
(38)  </attribute>
(39)  <attribute>
```



```

(40)      <description>
(41)          Define el nombre de la columna de la tabla.
(42)      </description>
(43)      <name>column</name>
(44)      <required>>false</required>
(45)      <rtexprvalue>>false</rtexprvalue>
(46)  </attribute>
(47)  <attribute>
(48)      <description>
(49)          Define el identificador del sistema de referencia espacial.
(50)      </description>
(51)      <name>srid</name>
(52)      <required>>true</required>
(53)      <rtexprvalue>>false</rtexprvalue>
(54)  </attribute>
(55) </tag>

```

Descripción

Entre las líneas (1) y (55) se declaran las características del *tag*. En la línea (3) se encuentra la descripción del *tag*, que se declara dentro de la etiqueta `<description>` (líneas (2) y (4)). En la línea (5) se define el nombre del *tag*, *geostd*, y en la línea (6) se define si el contenido tiene código Java. De la línea (7) a la (14) se define el atributo llamado *distance* en la línea (11) con su descripción en las líneas (8), (9) y (10), este atributo no es obligatorio al utilizar el *tag*, y se indica en la línea (12) y en la línea (13) se declara que no se evaluarán expresiones de otras JSP como valor de este atributo.

De la línea (15) a la (22) se declara el atributo llamado *gid*, línea (19), refiriéndose al identificador de objetos geográficos, este atributo se ocupa en varias consultas pero no es obligatorio, así lo indica la línea (20). La descripción está de la línea (16) a la (18), y no evalúa expresiones de otras JSP y se indica en la línea (21).

Los atributos llamados *table* (de la línea (23) a la (30)), *geotype* (de la línea 31 a la (38)) y *column* (de la línea 39 a la (46)) definen el nombre de una tabla, el tipo de objeto geográfico sobre el cual se consulta y el nombre de una columna de la tabla en la base de datos respectivamente. El atributo llamado *srid* (de la línea (47) a la (54)) define el identificador del sistema de referencia. La forma de leer la descripción de estos últimos atributos del *tag* es parecida a la de los primeros.

Ejemplo de utilización del *tag geostd*

Se presenta una parte de código para mostrar el uso del *tag geostd* para una consulta geográfica con lenguaje SQL:

```

(1)<c:set var="qryadd">
(2)  <tags:geostd type="addgeometrycolumn" table="user_locations"
(3)    geotype="POINT" srid="0"></tags:geostd>
(4)</c:set>
(5)  <sql:query sql="\${qryadd}" var="res" dataSource="\${bd}"></sql:query>

```

Descripción

En la línea (1) se inicia la etiqueta `<c:set>` de la biblioteca JSTL para establecer una variable, el atributo `var` es el nombre de la variable y el valor de la variable esta en las líneas (2) y (3), en donde se utiliza el `tag geostd` creado en este trabajo de tesis. En la línea (2) se inicia el `tag` llamado `<tags:geostd>` con los atributos `type`, `table`, `geotype` y `srid` y se termina en la línea (3). En la línea (5) se ocupa el `tag <sql:query>` de JSTL para realizar la consulta, en donde al atributo `sql` se le asigna la variable definida en nuestro `tag`. La consulta mostrada en el ejemplo declara una columna de tipo `geometry` en la tabla `user_locations` para el soporte de objetos geométricos.

Ejemplo de utilización del `tag geostd`

Se presenta una parte de código para mostrar el uso del `tag geostd` para insertar un punto geográfico que indique la localización de una persona.

```
(1)<sql:update var="qins" dataSource="{bd}">
(2) <tags:geostd type="insertgeometry2" table="user_locations"
(3)column1="5" column2="carlos" geometryobject="-3545.56 2341.35" srid="-1">
(4) </tags:geostd>
(5)</sql:update>
```

Descripción

En la línea (1) se inicia el `tag <sql:update>` de la biblioteca JSTL con los atributos `var` y `datasource` que son el nombre asignado a la etiqueta y el nombre de la base de datos respectivamente. En las líneas (2), (3) y (4) se asigna el contenido de la consulta, dado que el `tag` es `sql:update` sólo puede haber actualizaciones o inserciones en el contenido. En la línea (3) se usa nuestro `tag <tags:geostd>` para insertar la localización de un usuario a través de sus coordenadas geográficas. Entre sus atributos se declara el nombre del usuario y el sistema de referencia por defecto.

Estructura de un `tag` de la biblioteca GISWEBTAG

En la siguiente parte de código se presenta una JSP en el cual se hace uso de los `tags` creados en esta tesis. De la línea (1) a la (3) se toman los atributos que se envían como parámetros en la URL para que las JSP el servidor Web los interpreten, en este caso: `type`, `fid` y `distance`; de las línea (4) a la (7) se declaran las bibliotecas que se utilizarán en la JSP, la biblioteca JSTL y en la línea (7) se declara que se utilizarán los `tags` propios. En las líneas (8) y (9) se establecen variables para almacenar los parámetros recibidos, también se usa el lenguaje EL (*Expression Language*, en inglés) que permiten las JSP. Existen dos versiones de JSP, la 1.2 y la 2.0, la versión 2.0 es la que soporta el lenguaje EL. Desde la línea (10) y hasta la (40) se emplea la función `choose` de la biblioteca JSTL para comparar la variable `tipo` y para elegir la consulta adecuada en función de los parámetros. De la línea (11) a la (39) se encuentran las opciones según el tipo de consulta.

```
(1)<%@ attribute name="type" %>
(2)<%@ attribute name="fid"%>
(3)<%@ attribute name="distance"%>
```

```

(4)<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
(5)<%@ taglib prefix="sql" uri="http://java.sun.com/jstl/sql" %>
(6)<%@ taglib prefix="fn" uri="http://java.sun.com/jstl/fn" %>
(7)<%@ taglib prefix="tags" tagdir="/WEB-INF/tags" %>

(8)<c:set var="tipo" value="{type}"/>
(9)<c:set var="distancia" value="{distance}"/>

(10)<c:choose>
(11) <c:when test="{tipo == 'hidrologia'}">
(12)     <c:out value="SELECT AsGml(Centroid(the_geom))
(13)     FROM ra_referenciatopografica WHERE fid_ = {fid}"/>
(14) </c:when>
(15) <c:when test="{tipo == 'limites'}">
(16)     <c:out value="SELECT asgml(z.the_geom)
(17)     FROM ll_limites z,ng_toponimia c
(18)     WHERE c.id = (SELECT id FROM ng_toponimia WHERE
(19)     nom_ofi = {localidad} ) AND expand(makepoint((SELECT utm_x FROM
(20)     ng_toponimia WHERE id=(SELECT id FROM ng_toponimia WHERE
(21)     nom_ofi = {localidad} )),(SELECT utm_y FROM ng_toponimia WHERE
(22)     id=(SELECT id FROM ng_toponimia WHERE nom_ofi = {localidad} )),
(23)     3000) && z.the_geom AND distance(z.the_geom, makepoint((SELECT
(24)     utm_x FROM ng_toponimia WHERE id=(SELECT id FROM ng_toponimia
(25)     WHERE nom_ofi = {localidad} )),(SELECT utm_y FROM ng_toponimia
(26)     WHERE id=(SELECT id FROM ng_toponimia WHERE
(27)     nom_ofi = {localidad} )))) <= 3000"/>
(28) </c:when>
(29) <c:when test="{tipo == 'referenciaTopografica'}">
(30)     <c:out value="reftopografica"/>
(31) </c:when>
(32) <c:when test="{tipo == 'toponimia'}" >
(33)     <c:out value="SELECT assvg(z.the_geom) FROM {table1} z,
(34)     {table2} c WHERE c.id = {iden} AND expand(makepoint((SELECT
(35)     utm_x FROM {table2} WHERE id={iden})),(SELECT utm_y
(36)     FROM {table2} WHERE id={iden})), {distance}) && z.the_geom
(37)     AND distance(z.the_geom, makepoint((SELECT utm_x FROM {table2}
(38)     WHERE id={iden})),(SELECT utm_y FROM {table2} WHERE id={iden})))
(39)     <= {distance};" escapeXml="false"/>
(40) </c:when>

```

Uso del *framework Prototype* para la comunicación asíncrona

Prototype es un *framework* diseñado en *JavaScript* que tiene por objetivo de facilitar de desarrollo de aplicaciones Web dinámicas. El *framework Prototype* permite utilizar llamadas de forma asíncrona son la tecnología AJAX de una forma fácil. Además de peticiones

asíncronas simples permite tratar de una forma inteligente código *JavaScript* desde un servidor. En específico, en las pruebas realizadas, se hace uso de la clase `Ajax.Updater`, el cual ofrece la función de actualizar una página Web, un fragmento de la página o una parte del documento HTML mediante la actualización de alguno de sus componentes a través de su DOM (*Document Object Model* [26]).

```
(1)<head>
(2) <title>Hidrologia</title>
(3) <meta name="viewport" content="width= device-width"></meta>
(4) <link rel="style" type="text/css" media="screen" href="style.css"></link>
(5) <script type="text/javascript" src="lib/firebug/firebug.js"></script>
(6) <script type="text/javascript" src="lib/prototype-1.6.0.2.js"></script>
(7) <script type="text/javascript" src="codigo.js"></script>
(8) <link rel="stylesheet" href="RoundRectangle.css" />
(9) <jsp:include page="testgeo.jsp" flush="true" >
(10) <jsp:param name="ty" value="<%=objeto%>" />
(11) </jsp:include>
(12)</head>
(13)<script type="text/javascript" language="JavaScript">
(14) function getMap() {
(15) var g = document.getElementById("gid").value;
(16) new Ajax.Updater(
(17)     'topo',
(18)     'topoResponse.jsp?fid='+g,
(19)     'topo',
(20)     {asynchronous:true});
(21) }
(22)</script>
```

Requisitos e importación de la biblioteca de *tags* propia para su utilización en SIG

Para importar y utilizar la biblioteca de *tags* para la visualización de objetos geográficos en sus sistemas Web, son necesarios tres aspectos principales en la configuración previa al uso de los *tags* creados en este trabajo:

- Contar con un sistema administrador de base de datos espaciales *PostgreSQL / PostGIS*,
- obtener los datos en el sistema de coordenadas UTM,
- contemplar que todas las columnas que contengan atributos geométricos (tipo *geometry*) deben llamarse `the_geom`.

5.7. Modelado UML

Los requerimientos del sistema se detallan de una forma técnica mediante el lenguaje de modelado unificado (UML), lo que conlleva especificar el sistema en un conjunto de modelos, estos son representaciones gráficas que describen el problema a resolver y el sistema desarrollar. Los diagramas UML pueden representar el sistema desde diferentes perspectivas [58]:

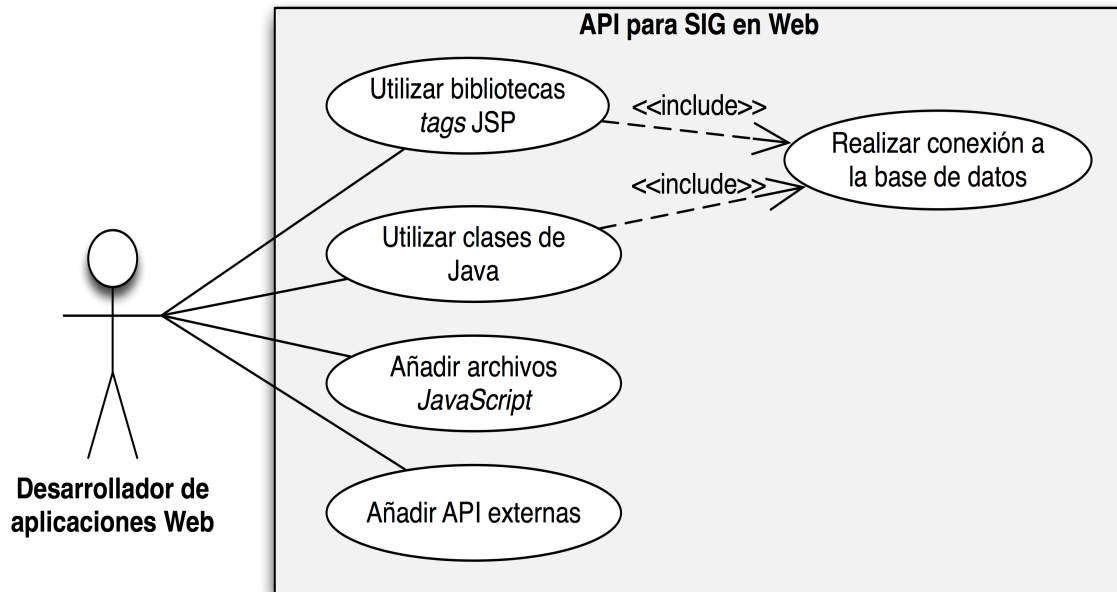
1. una perspectiva en la que se modela el contexto del sistema,
2. una perspectiva dinámica en la que se modela el comportamiento del sistema,
3. una perspectiva estructural en la que se modela la arquitectura del sistema o de la estructura de los datos procesados por éste.

Para documentar y analizar los requerimientos del proyecto se usó el modelado UML, a través de los diagramas que representan la estructura del sistema y otros que nos muestran el comportamiento dinámico e interacción entre los componentes de la API.

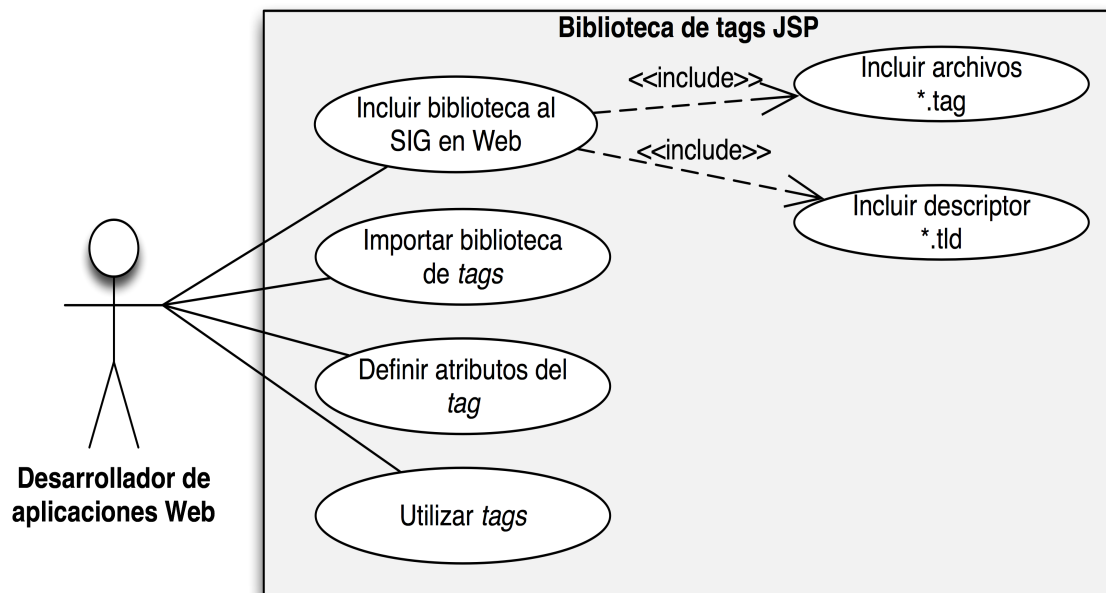
También se modela con el lenguaje UML para que se tenga una comprensión del problema a resolver, que en este caso es crear una API para desarrollar SIG en Web para dispositivos móviles.

Entre las tareas básicas que un desarrollador de aplicaciones Web puede realizar se definió que: es capaz de utilizar bibliotecas de *tags* JSP para manejar funciones, como realizar ciclos o condiciones, y realizar conexiones a sistemas de bases de datos; puede utilizar clases de Java (*JavaBeans*); importar código *JavaScript* y hacer uso de API externas. En específico, para usar la biblioteca de *tags* en la construcción de sistemas Web, existen tareas que el desarrollador puede realizar, como son: el incluir la biblioteca en la aplicación Web dentro del servidor, lo que incluye importar los archivos con extensión `*.tag` y el descriptor con extensión `*.tld`. Otra forma de compartir los *tags* es empaquetarlos en un archivo `*.jar`, en su caso, este archivo se debe incluir en la carpeta `lib` de la aplicación Web.

En la Figura 5.6, se muestran diagramas de caso de uso para obtener una perspectiva desde el punto de vista de un desarrollador de SIG en Web. En Figura 5.6-(a) se muestra un caso de uso general con las acciones que puede utilizar el desarrollador. En la Figura 5.6-(b) está el caso de uso cuando el desarrollador usa la biblioteca de *tags* propia.



(a)



(b)

Figura 5.6: Diagrama de caso de uso del desarrollador de aplicaciones Web GIS

Para conocer el comportamiento de los datos en el sistema se utilizaron modelos de comportamiento. También modelan la forma en que reaccionan los componentes del sistema a los eventos provocados por el usuario desde el dispositivo móvil. El usuario selecciona un elemento, entonces la aplicación Web recupera la información para enviarla al servidor y realizar la consulta al sistema gestor de la base de datos. Después, el servidor Web a través de *tags* transforma el resultado de la consulta geográfica a un formato gráfico para ser incrustado en el documento HTML. Luego, por medio del navegador Web se muestran los objetos al usuario.

En la Figura 5.7 se muestra un diagrama de actividad el cual hace énfasis en las actividades y las responsabilidades de los elementos que interactúan en el sistema (**Usuario**, **Aplicación cliente** y **Servidor de mapas**).

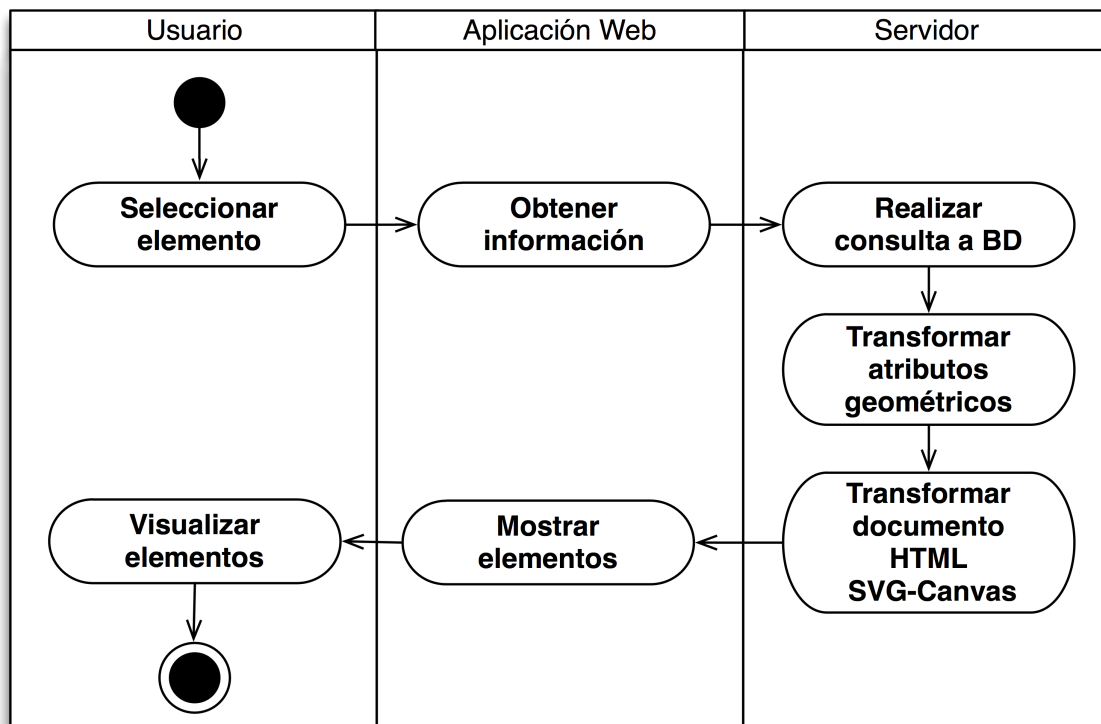


Figura 5.7: Diagrama de actividad de la generación del objeto geográfico

Los elementos técnicos que actúan en el proceso para visualizar los objetos geográficos en el navegador Web del dispositivo móvil son: el navegador Web, el objeto *XMLHttpRequest*, el servidor web y el sistema gestor de bases de datos. El usuario del SIG en Web genera eventos que son detectados por el navegador mediante código *JavaScript* en los elementos de la página, entonces el objeto *XMLHttpRequest* realiza la petición al servidor Web y éste último crea una instancia de una conexión al sistema de bases de datos. La comunicación que genera el objeto *XMLHttpRequest* es por medio de mensajes con estructura XML. Después de tener el resultado de la consulta se incrusta el gráfico en la página y se muestran los objetos geográficos y se pueden visualizar los mapas.

La Figura 5.8 muestra un diagrama de secuencia para complementar el modelo de comportamiento que se espera en el sistema. Éste se enfoca en el comportamiento dinámico de los elementos del sistema y como interactúan a través del tiempo, es decir, se muestran los elementos participantes en detalle (**WebGIS, ObjetoAjax, Servidor / JSP y Base de datos**) y el tiempo en el cual envían un mensaje a otro elemento, ofreciendo la funcionalidad al sistema.

En específico, para la aplicación cliente propuesta y por ser diseñada para Web 2.0 / AJAX se toma en cuenta la asincronía del objeto AJAX¹⁰.

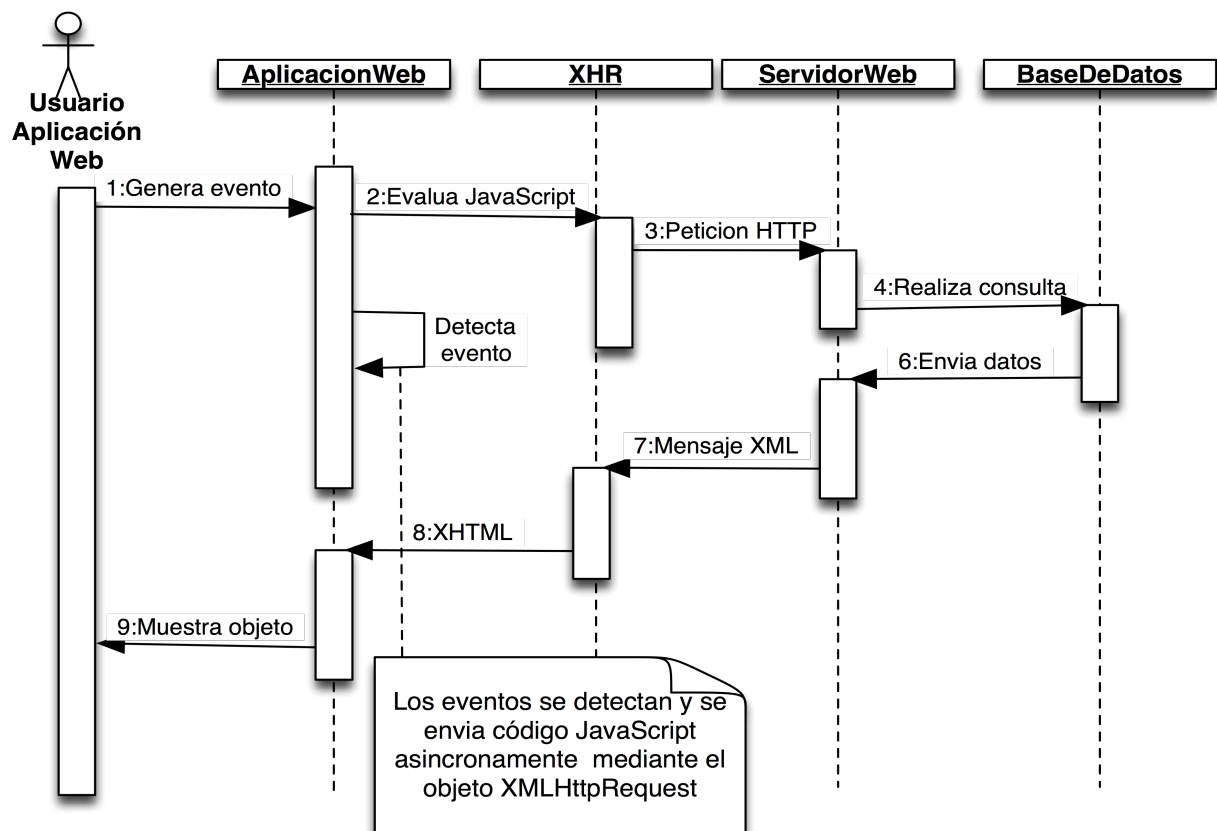


Figura 5.8: Diagrama de secuencia de la generación del objeto geográfico

¹⁰se refiere al uso del objeto *XMLHttpRequest* de tecnologías Web AJAX

Según Yeung y Hall [36] los subsistemas de un SIG en general son: el de entrada de datos, el de almacenamiento y consultas, el de manipulación y análisis, y el de visualización. En donde el subsistema de almacenamiento y consultas, realiza las actividades de edición y actualización de datos geográficos, además de consultar datos geográficos; el subsistema de manipulación y análisis realiza las actividades de agregar datos, eliminar datos y estimar parámetros; y el subsistema de visualización realiza el despliegue de datos, y los presenta de forma tabular, gráfica y en mapas.

Para modelar de forma genérica un sistema de información geográfica en Web se presenta en la Figura 5.9 el diagrama de actividades para un SIG para mostrar el contexto del trabajo de tesis, ya que ésta se enfoca en el subsistema de visualización y en específico para dispositivos móviles.

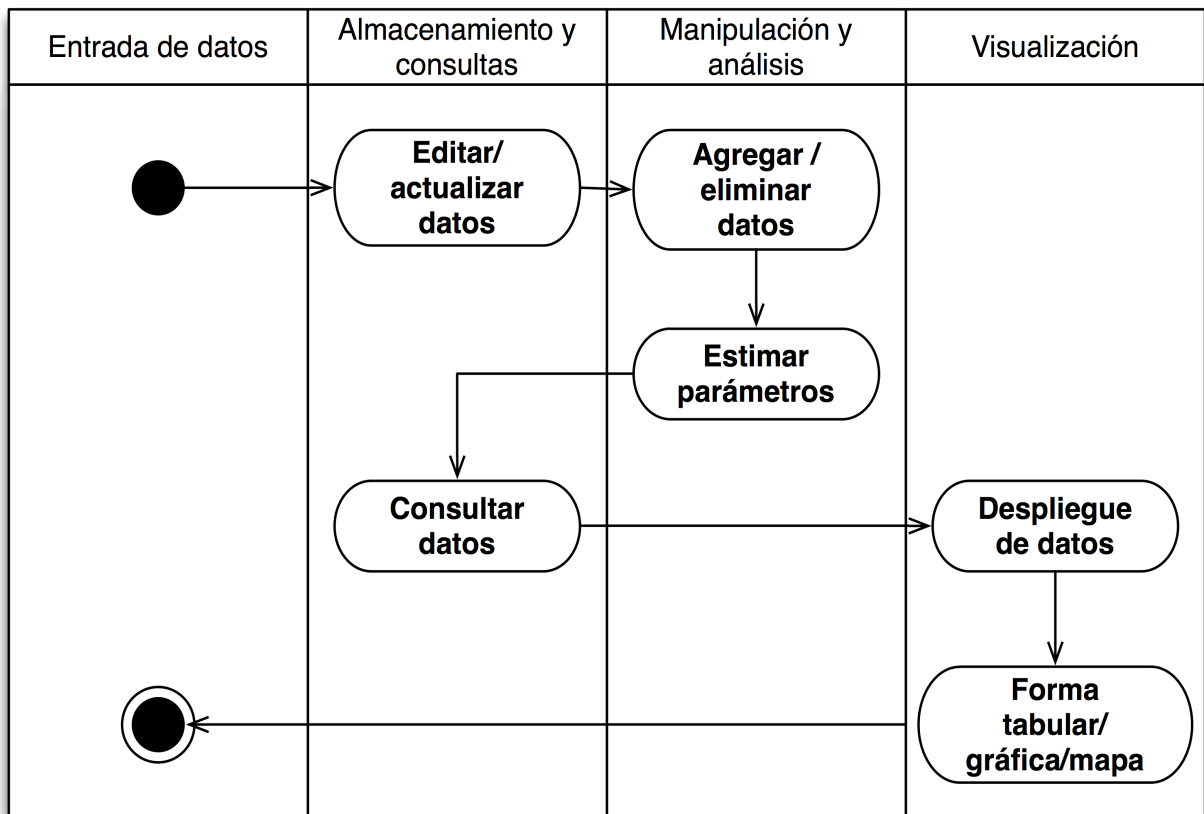


Figura 5.9: Diagrama de actividades de un SIG en Web

Siguiendo los conceptos de Yeung y Hall [36], se encuentra una dependencia entre los subsistemas. El sistema general está formado por los subsistemas de entrada de datos y preprocesamiento; almacenamiento y consultas; manipulación y análisis; y visualización en navegador Web. De la misma forma, el subsistema de almacenamiento y consultas depende de los sistemas gestores de bases de datos.

Estas dependencias entre subsistemas se pueden plasmar mediante un diagrama de componentes. En la Figura 5.10 se muestran los subsistemas, como componentes. El componente del sistema de visualización en el navegador Web se le asigna un estereotipo de «application» y al componente del sistema de bases de datos se le asigna el estereotipo «database». Los demás componentes son con estereotipo «system».

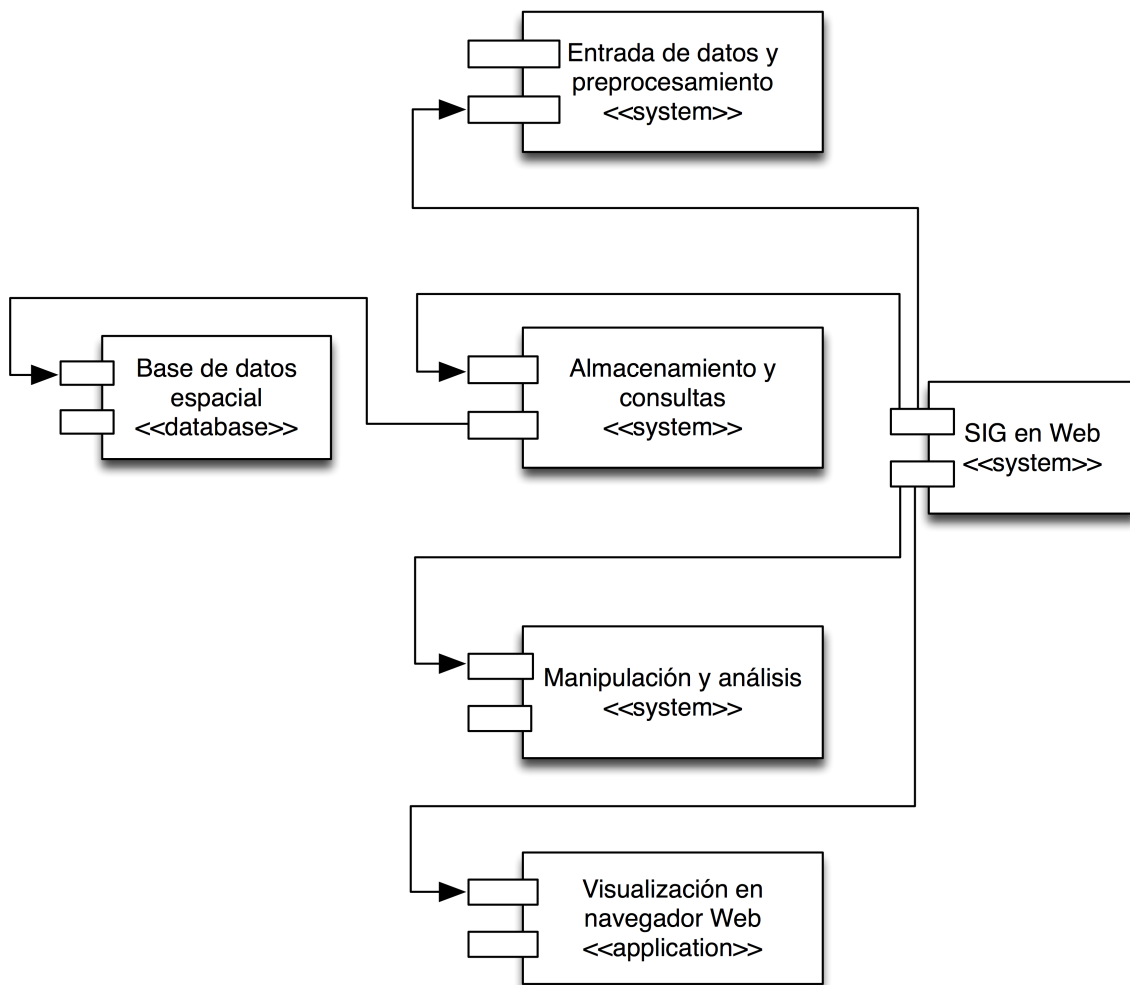


Figura 5.10: Diagrama de componentes de un SIG en Web genérico

Resumen del capítulo

En este capítulo se presentó el diseño de nuestra propuesta de solución. Primero se explicó la estructura que tiene la API con un enfoque para el desarrollo para la reutilización de componentes y la metodología para lograrlo. Después se analizaron los componentes, que son: la biblioteca de *tags* JSP, en donde se detallan las ventajas de usar *tags* (*e.g.* no es necesario que el desarrollador que ocupe la API sea experto en Java, sólo es necesario saber HTML), se describe la estructura de un *tag* y cada uno de los elementos que lo forman. También, se explicó el proceso para generar gráficos en formato Canvas y SVG, como se definieron los requerimientos para utilizar la API. Para validar nuestro trabajo se presentan diseños de SIG, el primero es un SIG en Web para el desarrollo sustentable en hidrología, que fue presentado en la sesión de pósters en el congreso WWDC'08 de Apple (*World Wide Developer Conference*) y el segundo un SIG en Web para la toma de decisiones en la infraestructura educativa.

Después, se definieron, describieron y ejemplificaron los *tags* creados en este trabajo, describiendo parte de los códigos generados para la creación de la biblioteca. Entonces, se modeló la biblioteca de *tags* JSP como componente esencial. Se utilizó el lenguaje UML para modelar, creando casos de uso, diagramas de actividad, de secuencia y de componentes.

En el próximo capítulo se muestran las validaciones a la biblioteca desarrollada con la creación de dos sistemas de información geográfica sobre el paradigma Web 2.0 y para los dispositivos móviles *iPodTouch* / *iPhone*.

Capítulo 6

Validación de SIG Web utilizando la biblioteca GISWEBTAG

En este capítulo se describen las pruebas que se realizaron para la validación de los componentes desarrollados. En la primera parte se muestran validaciones básicas de la generación de mapas mediante la biblioteca de *tags* JSP que se creó llamada GISWEBTAG. En las secciones siguientes se describen los sistemas que se crearon para la validación.

La primera validación es un SIG Web para el desarrollo sustentable en hidrología el cual utiliza datos espaciales del estado de Colima para las pruebas. Además mezcla, los SIG Web con API de otros sistemas como el de Mapas de Google (*Google Maps*) [3] y el de creación de gráficos de Google (*Google Chart*) [67]. También enlaza el sistema Web a la aplicación para el envío de correos electrónicos llamada *Mail* de los dispositivos iPod-Touch/iPhone.

La segunda validación es un SIG Web para la toma de decisiones en la infraestructura educativa de México, en éste se realiza la generación de objetos geográficos trascentes en ciertas regiones geográficas con cercanía a las escuelas, por ejemplo la capa de los rasgos hidrológicos cercanos a las escuelas o bien la capa de las vías de comunicación.

6.1. Validaciones básicas

En esta sección se presentan diversas pruebas para validar la biblioteca GISWEBTAG. Entre otros aspectos se presentan imágenes de diferentes capas haciendo la comparación con la imagen digital de la carta topográfica (extensión *gif* o *tif*). Cabe mencionar que se presentan partes de código para demostrar la sencillez con la que se pueden obtener gráficos generados automáticamente.

En la siguiente parte de código se muestra el uso de los *tags* propios que regresan la consulta en lenguaje SQL, formada dependiendo de los valores en los atributos de la etiqueta. El *tag* utilizado se llama *geoqueries*. También se hace uso de la biblioteca JSTL (*Java Server Pages Standard Tag Library*) para funciones estándar en *tags*.

```
(1) <c:set var="id_obj">61</c:set>
(2) <c:set var="qry">
(3) <tags:geoqueries type="toponimia" fid="{id_obj}"
(4)   table1="vt_viasdetransportacion"
(5)   table2="ng_toponimia" distance="3000">
(6) </tags:geoqueries>
(7) </c:set>
(8) <c:set var="qry1">
(9) <tags:geoqueries type="toponimia" fid="{id_obj}"
(10)  table1="cn_curvasdenivel"
(11)  table2="ng_toponimia" distance="3000">
(12) </tags:geoqueries>
(13) </c:set>
(14) <c:set var="qry2">
(15) <tags:geoqueries type="toponimia" fid="{id_obj}"
(16)  table1="hl_corrientesdeagua"
(17)  table2="ng_toponimia" distance="3000">
(18) </tags:geoqueries>
(19) </c:set>
(20) <c:set var="qry3">
(21) <tags:geoqueries type="toponimia" fid="{id_obj}"
(22)  table1="ra_referenciatopografica"
(23)  table2="ng_toponimia" distance="3000">
(24) </tags:geoqueries>
(25) </c:set>
```

Descripción

En la línea (1) se establece el identificador de la zona geográfica en la variable llamada `id_obj`. De la línea (2) a la (7) se define la variable `qry`, a la que con ayuda del *tag* propio `geoqueries`, línea (3), se le asigna la consulta SQL geográfica para obtener la toponimia y las vías de transportación en un radio de 3 kilómetros, que se asigna en la línea (5), y con centro en el objeto geográfico con identificador 61.

De la línea (8) a la (13) se declara una variable llamada `qry1`, a la que se le asigna una consulta SQL geográfica para mostrar las curvas de nivel, definido en la línea (10), que se encuentren en un radio de 3 kilómetros del objeto geográfico definido en la línea (1).

De la línea (14) a la (25) se definen las consultas para obtener las corrientes de agua (línea (16)) y la referencia topográfica (línea (22)) en un radio de 3 kilómetros del objeto geográfico definido en la línea (1).

Después de tener la consulta lista, se ejecuta en la base de datos y se regresa una variable que puede ser iterada con el *tag* `forEach` de JSTL para obtener los objetos geográficos, que están uno por cada fila de la variable. En el siguiente código se muestra como se van generando cada uno de los objetos geográficos con una etiqueta llamada `path` para su visualización.

```
(1) <c:forEach var="row" items="${results.rows}">
(2)     <svg:path d="${row.assvg}" stroke="black" fill="none"
(3)         stroke-width="15">
(4)     </svg:path>
(5)     <c:set var="i" value="${i+1}"></c:set>
(6) </c:forEach>
```

Descripción

En la línea (1) comienza el *tag* `forEach` con el cual se itera la variable que contiene el resultado de la consulta, en este ejemplo se llama `results`. En la línea (2) se comienza el *tag* para graficar polígonos en formato SVG llamado `<svg:path>`, se asignan en su atributo `d`, los datos a graficar, que son los que están en la columna llamada `assvg`. Los atributos restantes son `stroke`, `fill` y `stroke-width`, que definen el color, si es con relleno y el ancho de las líneas, respectivamente.

Para mostrar en detalle la declaración de un gráfico en formato SVG y cómo se insertan las coordenadas, se presenta la siguiente parte de código. Se utiliza la etiqueta *path* para generar polígonos o multilíneas.

```
(1)<svg:path d="M 633476 -2126426 633490 -2126445 633546
(2) -2126475 633615 -2126499 633678 -2126514 633730 -2126522
(3) 633762 -2126525 633778 -2126517 633874 -2126655 "
(4) stroke="black" fill="none" stroke-width="15">
(5) </svg:path>
```

Descripción

En la línea (1) se comienza la declaración del gráfico con la inserción de las coordenadas. Al iniciar cada objeto geográfico, en el atributo (`d`) se coloca la letra *M* antes de las coordenadas, esto significa *moveTo*, que indica que el objeto geográfico iniciará desde ese punto. Las líneas (2) y (3) contienen más coordenadas. Y la línea (4) define el color, si es con relleno y el ancho de la línea.

Uso del dispositivo móvil para realizar acercamientos y alejamientos

Se ocupa el recurso de la pantalla sensible al tacto del dispositivo móvil *iPodTouch/iPhone* para realizar un acercamiento (*zoom in*), generando un evento con los dedos sobre la pantalla táctil. La acción que se debe realizar, es tocar con dos dedos en puntos cercanos la pantalla táctil y sin levantarlos se deben mover a hacia la parte exterior, veáse la Figura 6.1. Para realizar un alejamiento completo del mapa se toca dos veces la pantalla rápidamente, veáse Figura 6.2. Cabe mencionar que los datos en el mapa no se actualizan al realizar los acercamientos, los datos se cargan en el navegador desde que se realiza la consulta espacial.

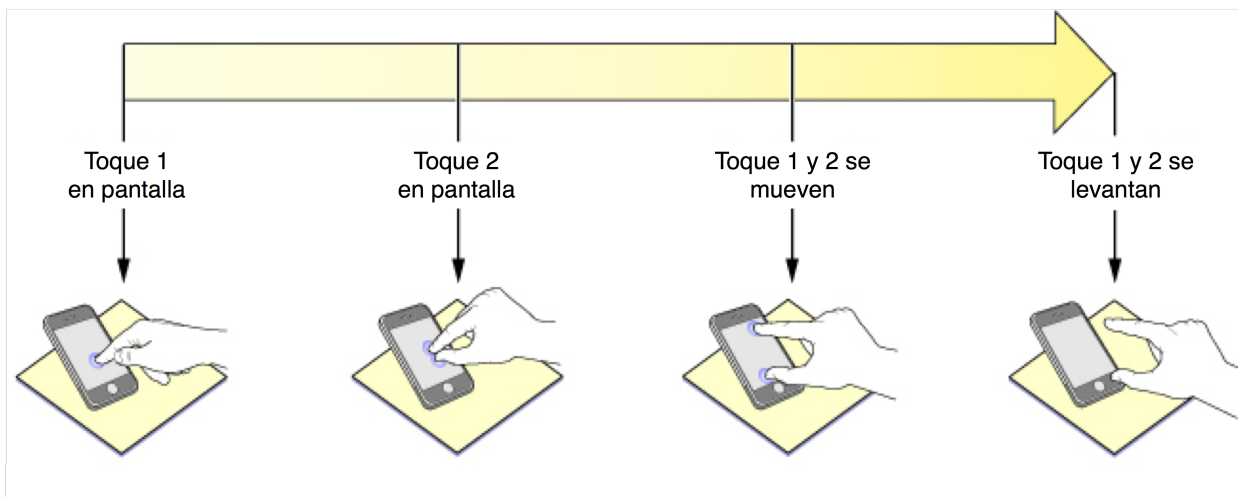


Figura 6.1: Flujo de movimientos para acercar los objetos en la pantalla sensible al tacto del dispositivo móvil

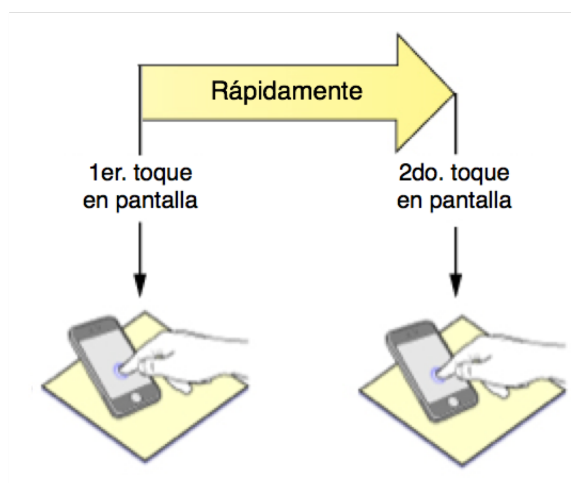


Figura 6.2: Flujo de movimientos para alejar completamente la vista de los objetos geográficos

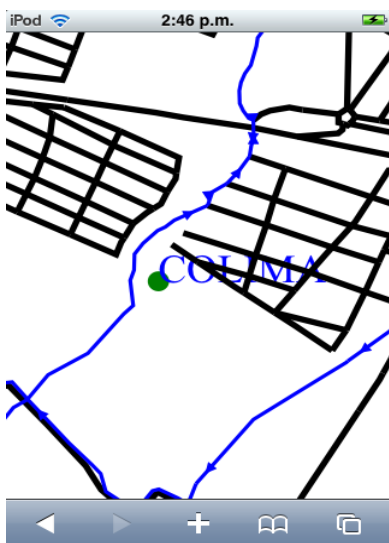
Para mostrar la resolución que se logra con el acercamiento en imágenes vectoriales en el dispositivo móvil se presentan imágenes de prueba con los mapas del estado de Colima. En la Figura 6.3-(a) se muestra la primer vista que se genera después de realizar la consulta, en la Figura 6.3-(b) se presenta una imagen del navegador Web cuando se realiza un acercamiento mediante el proceso mostrado en la Figura 6.1. En la Figura 6.3-(c) se visualiza cuando se han realizado todos los acercamientos posibles y en la Figura 6.3-(d) aparece la zona correspondiente de la carta topográfica.



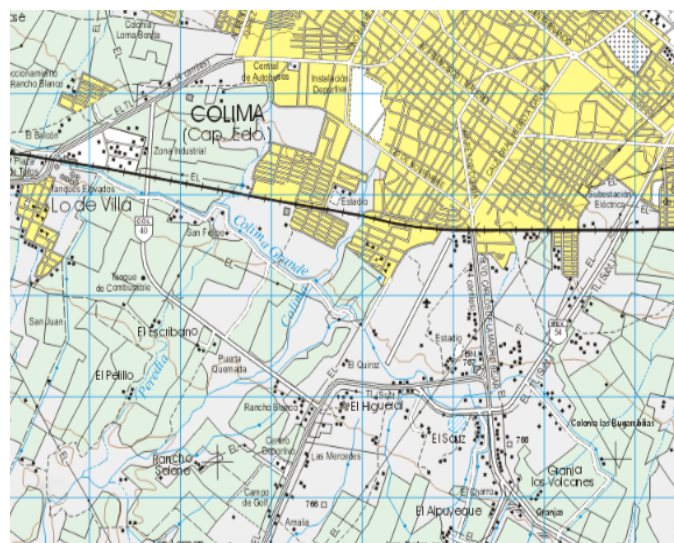
a)



b)



c)



d)

Figura 6.3: Imágenes de gráficos SVG en el navegador Web Safari del *iPodTouch*, acercamiento y comparación vs. la carta topográfica.

Las capa que corresponde a las vías de comunicación contiene las carreteras estatales y federales, la referencia topográfica define las áreas de los terrenos y las curvas de nivel son líneas que unen puntos de igual altitud sobre un terreno. En la Figura 6.4 se muestran estas capas en la pantalla del dispositivo móvil iPodTouch.

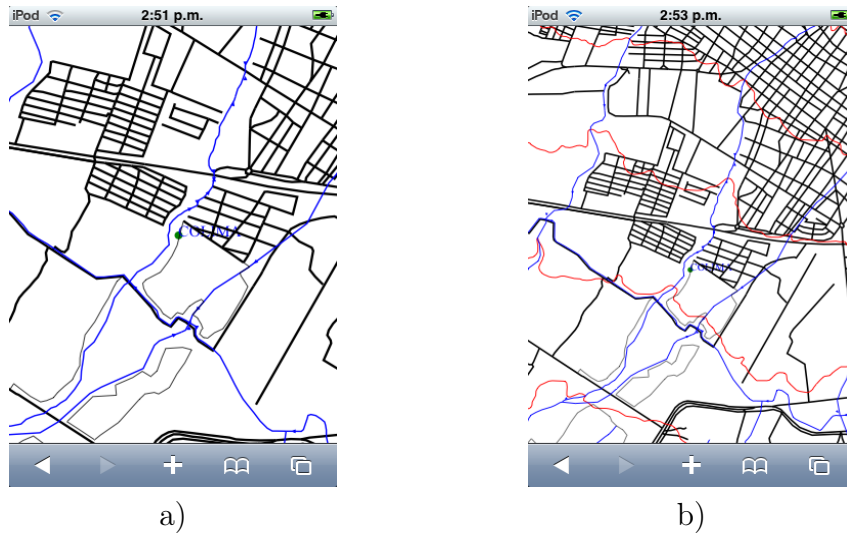


Figura 6.4: Gráficos en formato SVG, a) muestra la capa de las corrientes de agua (azul) vías de comunicación (negro) y referencia topográfica (cafe); b) además muestra las curvas de nivel (rojo).

En la Figura 6.5 se compara una zona montañosa mediante las curvas de nivel obtenida con formato SVG *vs* la carta topográfica de INEGI. En la Figura 6.6 se muestran diversas capas en SVG y la misma zona de la imagen en formato TIF, lo anterior para enfatizar la ventaja en la resolución que ofrece el formato gráfico SVG al ser una representación vectorial, en contra de representaciones en mapas de bits.

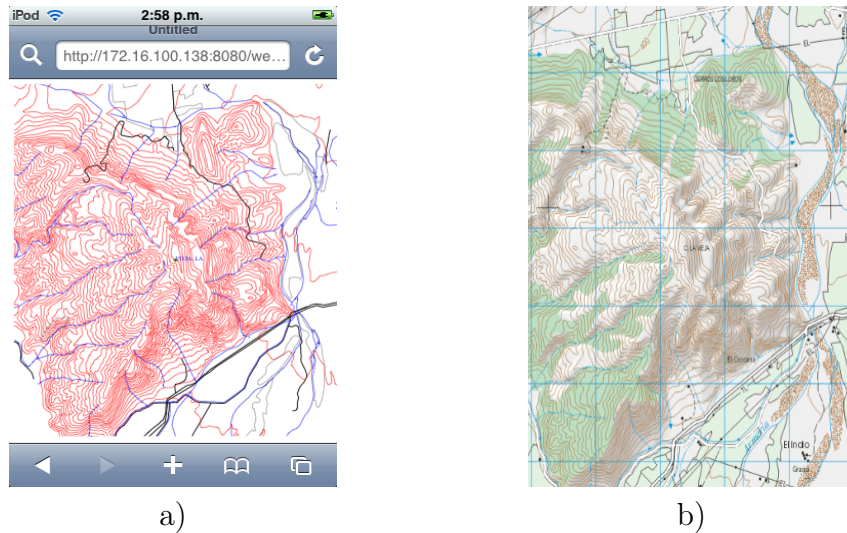


Figura 6.5: a) Visualización de las curvas de nivel de una zona montañosa, generada con SVG y b) es la carta topográfica de la zona.

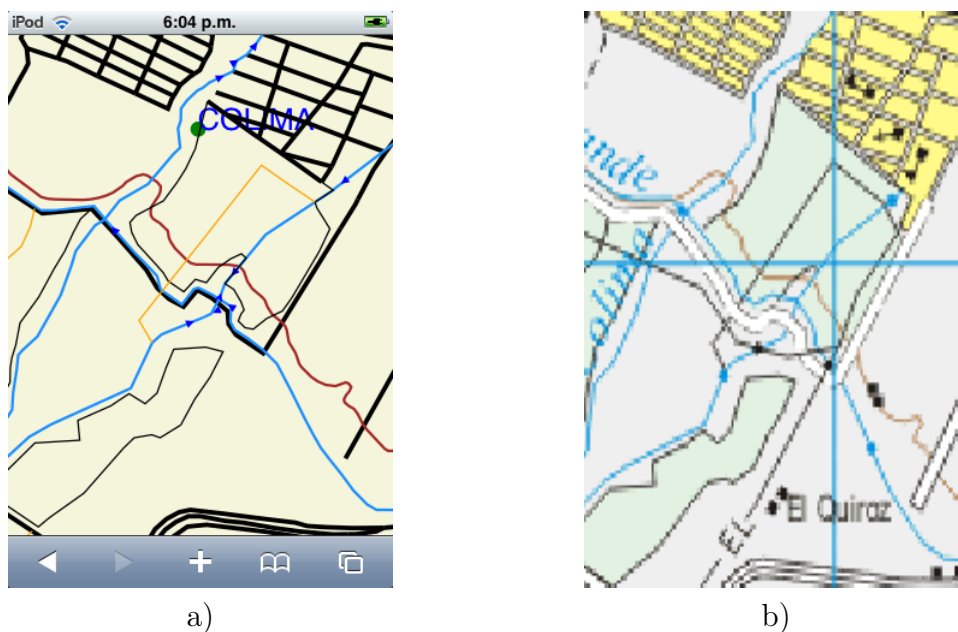
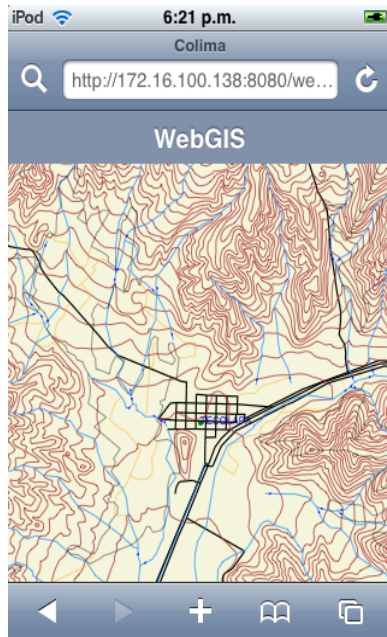
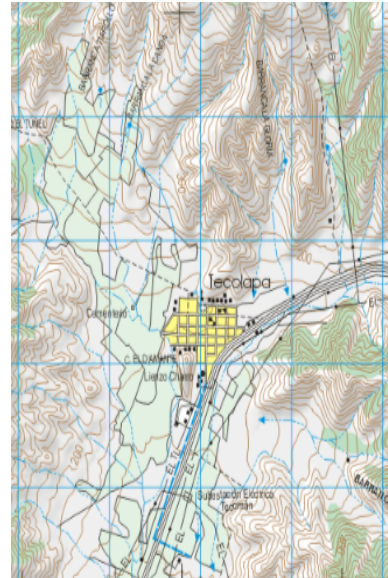


Figura 6.6: Visualización de todas las capas en SVG, mostrando la resolución en: a) formato SVG y b) imagen TIF.

En la Figura 6.7 se muestran las curvas de nivel, vías de comunicación y corrientes de agua dentro de la zona geográfica. Cabe mencionar que la capa de corrientes de agua, cada uno de los objetos geográficos en esta capa tiene un sentido que se muestra en el formato SVG con una flecha creada con objeto geométrico SVG.



a)



b)

Figura 6.7: Visualización de las capas de corrientes de agua (azul), curvas de nivel (cafe) y vías de transportación (negro).

6.2. Sistema Web de información geográfica para el desarrollo sustentable en hidrología

Para validar los componentes creados se creó este sistema Web que almacena dos tipos de datos: geográficos, que están dados por un conjunto de mapas; y climáticos como la temperatura, presión y humedad entre otros; las mediciones de estos últimos se recuperan usando sensores. Un módulo o nodo de sensores se conecta a un servidor para que se actualice la base de datos geográfica y las mediciones aparezcan en las próximas consultas.

El sistema combina bases de datos espacio temporal y bases de datos geográficas para lograr el objetivo del SIG, que es mejorar la regulación de los recursos hidrológicos en México. Sin embargo, debemos mencionar que solamente se realizó con datos del estado de Colima. La comunicación entre el servidor de bases de datos espacio temporal con el servidor Web es a través de la Internet y la comunicación entre el servidor y el cliente se realiza a través del acceso a la Internet desde de la red celular para el *iPhone* y a través de puntos de acceso inalámbricos 802.11 b/g para la *iPodTouch* e igualmente el *iPhone*. La arquitectura general de este proyecto se puede apreciar en la Figura 6.8.

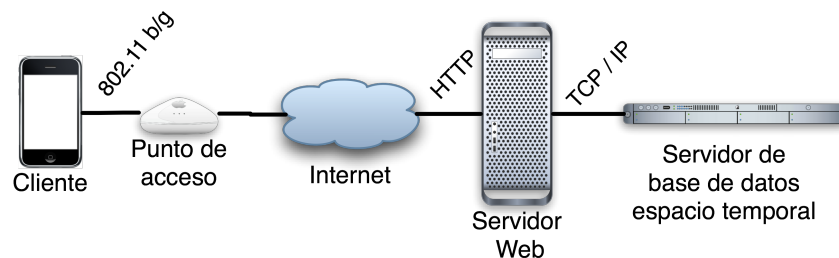


Figura 6.8: Arquitectura del sistema Web de información geográfica en Web para el desarrollo sustentable en hidrología.

Además, el SIG muestra una gráfica de los valores medidos por los sensores a través de tiempo con la ayuda de la API *Google Chart* [67]. Los nodos de medición son localizados estratégicamente en ríos, lagos o mares, estos nodos envían datos que son almacenados en la base de datos espacio temporal.

En la Figura 6.9 se muestra la localización de un nodo de sensores y los objetos geográficos que corresponden a la hidrología, en el momento de tocar en la pantalla el lugar en donde está el nodo aparece una alerta que indica el nombre del nodo, en el ejemplo el nodo se llama *node 2*. Después de la alerta aparece la información del nodo, el nombre del supervisor junto con su correo electrónico, dirección y sus números telefónicos (véase Figura 6.10).

Al utilizar los dispositivos móviles de Apple, se aprovechan las aplicaciones nativas de envío de correo electrónico (*Mail*) y de mapas (*Maps*). En donde aparece la información al tocar el texto *View Maps* realiza el enlace a la aplicación de Mapas que recibe las

coordenadas y se abre en mismo lugar que esta el nodo; al tocar en donde aparece el correo electrónico abre la aplicación de correo electrónico con el nombre de destinatario.

Cabe mencionar que este proyecto fue aceptado y presentado en el congreso de desarrolladores de Apple en la sesión de pósters de desarrollo científico. El congreso se llevó a cabo en San Francisco, California, en junio de 2008.

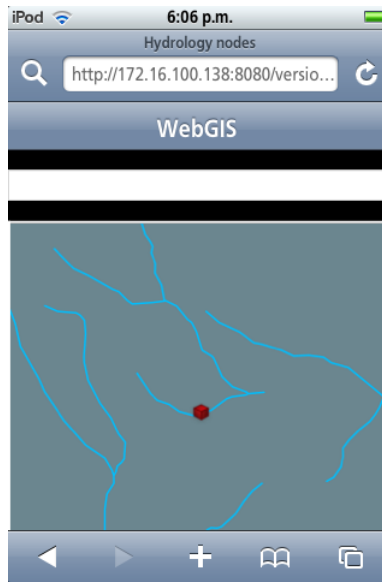


Figura 6.9: Mapa en donde se localiza un nodo de sensores

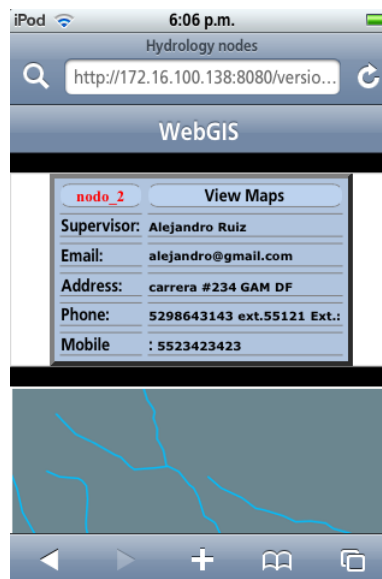
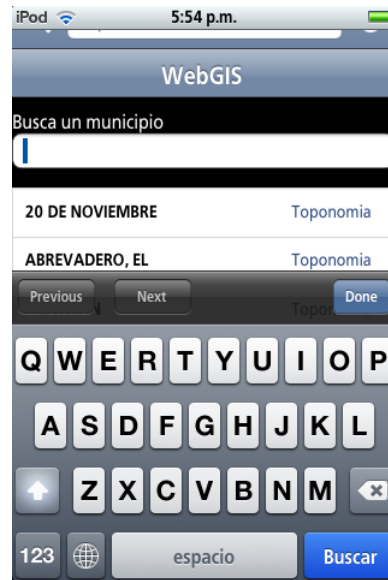


Figura 6.10: Información a detalle del nodo de sensores.

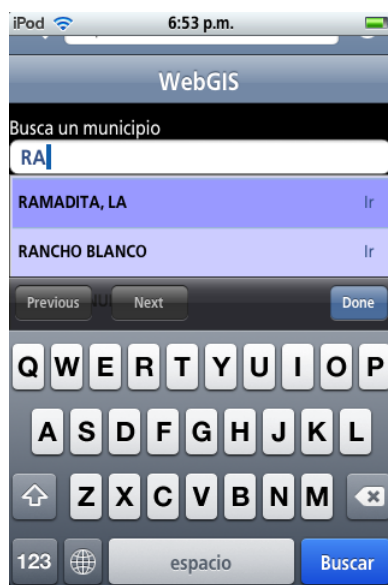
En la Figura 6.11 se muestran las formas de buscar lugares por su nombre, y debido a que los nombres son demasiados se incluye un campo para introducir el nombre del lugar



a)



b)



c)



d)

Figura 6.11: Proceso de búsqueda de localidades. a) Pantalla para iniciar búsqueda; b) y c) pantallas para mostrar como introducir nombre de localidad en el *iPodTouch*; d) pantalla mostrando los nombres de las localidades sugeridas de la base de datos.

para que aparezcan las sugerencias, lo anterior usando AJAX para realizar la conexión a la base de datos y el envío de datos de forma asíncrona.

En la Figura 6.12 se muestra el esquema conceptual de las tablas dentro de la base de datos para gestionar la información de los nodos de sensores. Recordando que el sistema gestor de las bases de datos es *postgreSQL/postGIS* el cual es del tipo relacional.

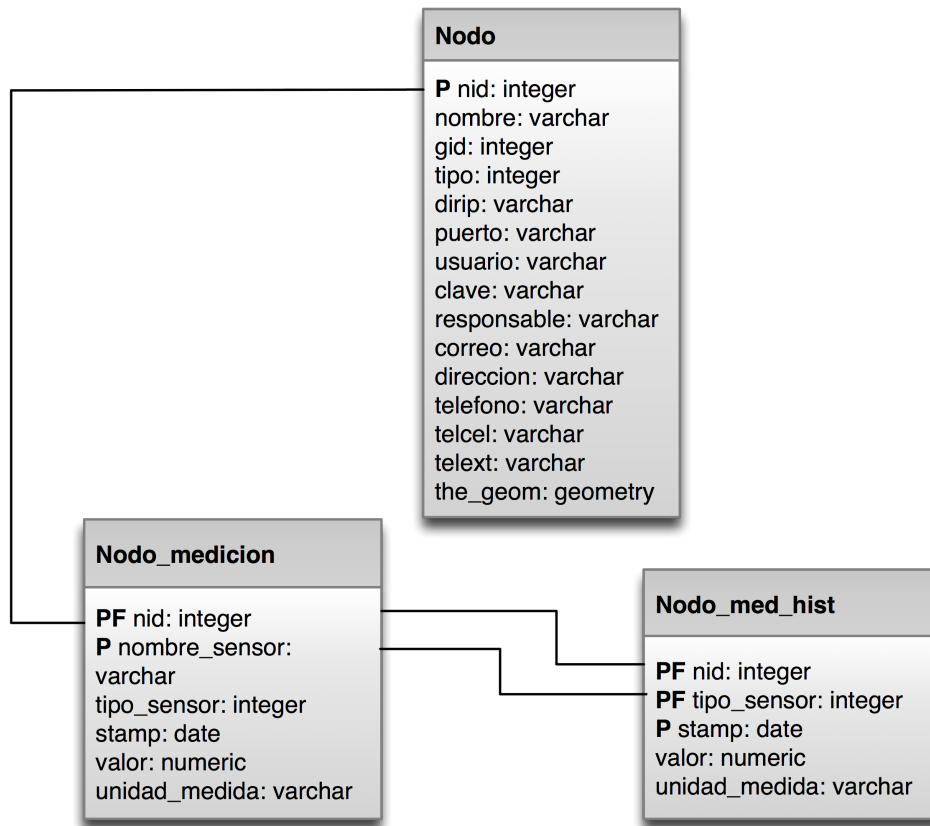


Figura 6.12: Esquema conceptual de las tablas para el caso de estudio del SIG Web para el desarrollo sustentable en hidrología.

6.3. Sistema Web de información geográfica para la toma de decisiones en la infraestructura educativa

Este sistema da soporte a la toma de decisiones para la planeación de infraestructura educativa, ya que permite visualizar lugares geográficos de las escuelas o institutos de educación. Así mismo permite mostrar objetos geográficos de la zona cercana a la escuela.

Principalmente, se almacenan datos espaciales de los objetos geográficos y de las escuelas en una base de datos espacial (véase Figura 6.13). Y por otra parte, se almacenan los atributos toponímicos correspondientes a las escuelas para conocer el nombre oficial de la escuela, dependencia, estado de la construcción; y algunos datos de la propia escuela como: nombre del director, número de estudiantes, turno y fecha de apertura.

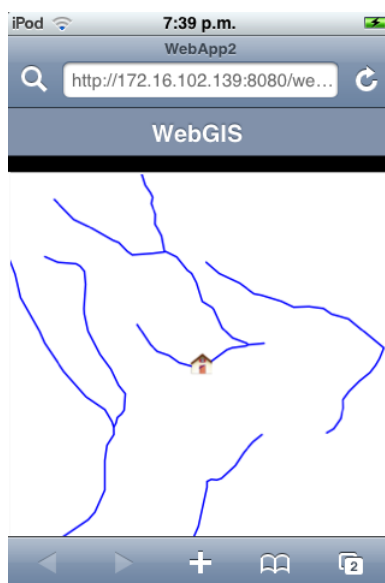


Figura 6.13: Captura de pantalla del sistema en donde se visualizan ríos y una escuela.

En la Figura 6.14 se muestra un acercamiento para el caso del formato Canvas, en donde hay limitaciones en la resolución. Cuando se toca la pantalla en la imagen de la “escuela” aparecen los datos de las escuela (nombre,) y su foto ocupando el *framework Scriptaculous* para insertar efectos durante el proceso.



Figura 6.14: En la Figura a) se muestra un acercamiento al mapa en formato Canvas e imagen cuando comienza a aparecer escuela y en b) su información.



Figura 6.15: Imagen de la escuela, datos toponímicos y del responsable.

En la Figura 6.15 se muestra la fotos completa de la escuela junto con sus datos, al tocar la pantalla sobre la imagen, se ocultará y se mostrará el mapa nuevamente. En la Figura 6.16 se muestra conjuntamente en a) la foto de la escuela y la localización de la escuela en el mapa. Además de mostrar en b) los datos de las escuela que se encuentran en la base de datos.



Figura 6.16: a) Pantalla que muestra el lugar geográfico y la foto de la escuela, b) Atributos de la escuela u objeto geográfico.

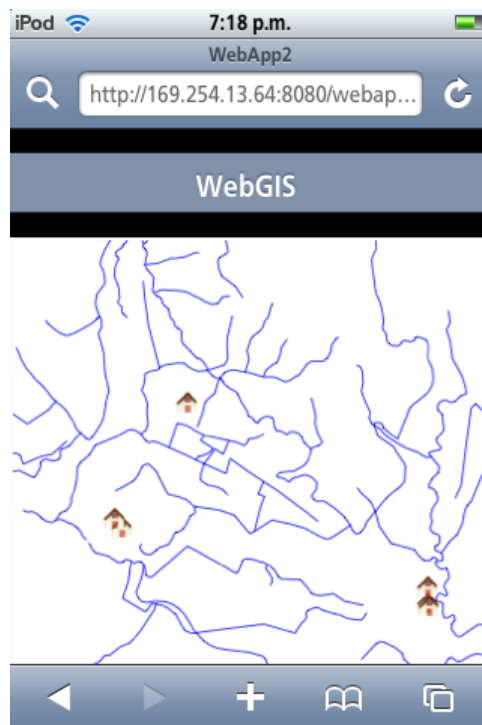


Figura 6.17: Región con diversas escuelas y la capa de hidrología.

En la Figura 6.17 se visualizan cinco escuelas en una zona con varias corrientes de agua, mostrando así que a partir de la visualización con diferentes capas se puede dar soporte para la planeación de infraestructura de escuelas.

Resumen del capítulo

En este capítulo se presentaron las validaciones que se realizaron a la biblioteca, y se muestran imágenes tomadas de la pantalla del *iPodTouch* de diversas capas, como la capa de las curvas de nivel o las vías de comunicación, y estas imágenes se comparan con las imágenes TIF de la carta topográfica. Por otra parte se describen dos SIG creados para validar la API de visualización, uno para el desarrollo sustentable y otro para apoyar la toma de decisiones en la infraestructura educativa y se muestran imágenes de algunas funciones del SIG.

En el próximo para culminar este trabajo se explican los resultados y conclusiones. Además de mencionar las contribuciones y el trabajo a futuro que sigue a mi consideración en esta área de investigación.

Capítulo 7

Conclusiones

7.1. Conclusiones

En general, podemos concluir la importancia que tiene el desarrollo de tecnología Web para los sistemas de información geográfica. La interconexión de grandes servidores de bases de datos geográficas con servidores Web cobra gran importancia dentro de la computación móvil [68]. El diseño y el desarrollo de una API, mostrado en esta tesis, es relevante para la visualización en el navegador Web Safari del dispositivo iPodTouch de Apple. Así, la biblioteca GISWEBTAG de *tags* JSP fue liberada (ver. 1.0) y se validó con dos casos de estudio de impacto social. El primero, para estudio del “Desarrollo sustentable de cuencas hidrológicas” para el estado de Colima¹. El segundo, corresponde a una propuesta para la ubicación de la infraestructura educativa y ayuda para la toma de decisiones de inversión.

7.1.1. Impacto tecnológico en Web

Una primera conclusión refiere a la importancia que tiene la generación dinámica de objetos geográficos en Canvas o en SVG, como una de las mejores opciones para desarrollar el subsistema de visualización SIG en Web. Las razones son la aportación de una mayor flexibilidad en la actualización del sistema y a los atributos de objetos geográficos. La visión completa de los objetos geográficos en su representación gráfica y nominal es un problema de interfaz hombre-máquina, que se resuelve con la visión del Canvas y los atributos al mismo tiempo o en forma flexible. Desde los 70'tas y 80'tas con los sistemas geográficos *Zeiss* de captura de información éste ha sido una tema heredado a los sistemas *standalone* y ahora en Web.

Se concluye, y es evidente que el desarrollo gráfico SVG² de formato vectorial es el que brinda una mejor resolución en los acercamientos sobre los mapas que los gráficos en *bitmap* o *raster*. Otra conclusión, concierne a la creación de la estructura SVG basada en la estructura XML. La importancia en el estándar SVG es que permite una inmediata actualización mediante el objeto *XMLHttpRequest*. De esta manera, podemos asegurar que SVG es el formato ideal para crear SIG en Web; por consiguiente, podemos crear

¹Proyecto SEP-CONACYT Project No. 61539

²El lector puede acudir a la página 81 de este trabajo.

aplicaciones enriquecidas para el usuario (RIA) con las tecnologías AJAX dentro del paradigma Web 2.0 [69].

7.1.2. Impacto en tecnología móvil

Con el auge de los dispositivos móviles para acceder a la Web se han creado lenguajes como:

- WML (*Wireless Markup Language*)
- XHTML-MP (XHTML en su versión para móviles , *Mobile Profile*)

Sin embargo, en mi opinión, hoy en día los futuros desarrollos de sistemas Web deben realizarse con mayor énfasis en las tecnologías Web (*e.g.* XML, CSS, XSLT, DOM, JavaScript). Con ésto, el objetivo es crear los sistemas Web sin importar el lenguaje de programación, debido a que los navegadores Web de los dispositivos móviles más recientes soportan gran cantidad de tecnologías de desarrollo Web.

7.2. Contribución

7.2.1. Contribución a los SIG

Desde los inicios de la cartografía tradicional con los sistemas de mapeo y captura, ofrecidos en gran medida con aparatos Zeiss, el problema de interacción hombre-máquina ha sido fundamental. El despliegue de mapas con señalamiento de coordenadas para introducir atributos nominales y guardarlos en cintas magnéticas fue uno de los problemas principales en el desarrollo de la cartografía.

Los SIG sustentados en grandes computadoras vinieron a presentar una primera revolución con los subsistemas de: entrada de datos; sistemas de adquisición y procesamiento de datos espaciales; almacenamiento y consultas; con sistemas de bases de datos geográficas; manipulación y análisis, con el sistema de visualización. En este momento la comparación de un SIG por computadora (*standalone*) *vs.* procesamiento de la cartografía tradicional mostraba grandes ventajas por el procesamiento y almacenamiento de los datos geográficos vectoriales y los atributos nominales, con el objetivo de ser modelados y llevados a la tecnología de bases de datos. En una nueva generación, el problema que enfrentamos es todo un sistema distribuidos y concurrente.

7.2.2. Contribución a los OGIS (*Open Geographic Information Systems*)

Es de gran importancia apearse a las especificaciones del OGC (*Open GIS Consortium*) para que los SIG en Web puedan incorporarse a otros SIG y los datos geométricos tengan el mismo estándar. La infraestructura utilizadas fue sistema gestor de bases de datos relacional *PostgreSQL*, con su aditamento para el soporte a datos espaciales llamado *PostGIS*, el cual maneja tipos de datos geométricos, además, que contiene funciones para

consultar los datos geométricos o espaciales. La principal contribución a los OGIS es que la API creada en este trabajo utiliza tecnologías abiertas (*OpenSource*) y puede ser implantada en SIG cumpliendo con las especificaciones del OGC.

7.2.3. Contribución tecnológica al desarrollo sustentable e impacto social

Se desarrollaron dos validaciones para visualizar mapas con impacto social, la primera un SIG en Web para el desarrollo sustentable en hidrología que fue presentado en el congreso de desarrolladores de Apple (WWDC'08) en San Francisco, California. La segunda, fue un sistema de información geográfica para la toma de decisiones en la infraestructura educativa de México.

7.3. Trabajo a futuro

El punto de partida para este trabajo, es el trabajo realizado en el laboratorio de super cómputo del departamento de computación del CINVESTAV, unidad Zacatenco. En específico, el trabajo de Hernandez [10] trata sobre la creación de SIG en Web de forma dinámica, y menciona como trabajo futuro, la creación de herramientas para desarrollar SIG en Web para dispositivos móviles, aspecto que se cubrió en mi proyecto de tesis en lo que corresponde a la visualización de objetos geográficos en el navegador Web del dispositivo móvil.

Desde mi punto de vista, este trabajo abre áreas de oportunidad en el desarrollo de sistemas Web de información geográfica, tanto en sistemas para navegadores Web convencionales como navegadores Web para dispositivos móviles. A continuación se mencionan las áreas a seguir a partir de este proyecto.

En tecnologías Web:

- Ampliar la API para que sea utilizado en otros ambientes de desarrollo en Web que no sea Java / JSP. Algunos ejemplos son: PHP, *Ruby on rails*, *GeoDjango*, entre otros.
- Desarrollar un sistema Web a través del cual se puedan administrar otros SIG en Web. Teniendo las ventajas de que el creador de estos sitios no tiene que ser experto en sistemas de bases de datos espaciales.

en Sistemas de Información geográfica:

- Crear las consultas espaciales de acuerdo al lenguaje para sistemas de bases de datos como *Oracle Spatial*, *DB Spatial* y *SQL Server spatial*.
- Crear aplicaciones (software) para sistemas de información geográfica en dispositivos móviles para la adquisición de datos geograficos en campo a través de un dispositivo GPS.

- Crear un sistema de información para administrar los datos espaciales que se encuentran en diferentes sistemas geográficos de referencia.
- Diseñar e implementar un módulo para realizar minería de datos sobre los datos espaciales. Como parte del subsistema de análisis de datos que tienen los SIG.

y en tecnologías para móviles.

- Crear una aplicación para el *iPhone* de Apple, con el fin de ocupar el acelerómetro interno para indicar la dirección a la que una persona desea dirigirse y el GPS para mostrar la ubicación del dispositivo.
- Realizar un servidor de mapas, a través del cual los usuarios puedan obtener y utilizar los datos espaciales en SIG para móviles.

Bibliografía

- [1] Map quest (2008, diciembre). disponible en: <http://www.mapquest.com>.
- [2] Nasa world wind (2008, diciembre). disponible en: <http://worldwind.arc.nasa.gov/>.
- [3] Google maps api (2008, diciembre). disponible en: <http://maps.google.com>.
- [4] Toshihiko Yamakami. Mobileweb 2.0: Lessons from web 2.0 and past mobile internet development. *Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on*, pages 886–890, April 2007.
- [5] Karen Church, Barry Smyth, Paul Cotter, and Keith Bradley. Mobile information access: A study of emerging search behavior on the mobile internet. *ACM Trans. Web*, 1(1):4, 2007.
- [6] Map accel (2008, diciembre). disponible en: <http://www.territoriumonline.com/en/gis/mapaccel/webgis.html>.
- [7] ESRI. Arcgis (2008, diciembre). disponible en: <http://www.esri.com/software/arcgis>.
- [8] Django Software Foundation. Geodjango (2008, diciembre). disponible en: <http://geodjango.org/>.
- [9] S. Manca, P. Cau, E. Bonomi, and A. Mazzella. The datacrossing dss: A data-grid based decision support system for groundwater management. *e-Science and Grid Computing, 2006. e-Science '06. Second IEEE International Conference on*, pages 107–107, Dec. 2006.
- [10] Claudia G. Hernández Pérez. Programación automática de sistemas web a partir de descripciones xml. caso de estudio: Sistemas de información geográfica. Master's thesis, CINVESTAV, 2008.
- [11] Michael DeMers. *Elements of cartography*. John Wiley and Sons Inc., New York, 2005.
- [12] A.H. Robinson-J.L. Morrison-P.C. Muehrcke-A.J. Kimerling and S.C. Guphill. *Fundamentals of geographic information systems*. John Wiley and Sons Inc., Hoboken, NJ, 1995.

- [13] D.F. Marble. The potential methodological impact of geographic information systems on the social sciences. *In interpreting space: GIS and archaeology*, 1990.
- [14] D.J. Coleman-P.A. Longley-M.F. Gooch-D.J. Maguire and D.W. Rhind. *Geographical Information Systems in Networked Enviroments*. John Wiley and Sons Inc., New York, 1999.
- [15] Z.R. Peng and M. H. Tsou. *Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks*. John Wiley and Sons Inc., Hoboken, NJ, 2003.
- [16] Tom O'Reilly. *What is web 2.0?* O'Reilly Media, 2005.
- [17] J. James Garrett. *AJAX: a new approach to Web applications* (2008, diciembre). disponible en: <http://www.adaptivepath.com/ideas/essays/archives/000385.php>.
- [18] J. Fawcett N. Zakas, J. McPeak. *Professional Ajax*. Wiley Publishing Inc., 2006.
- [19] Richard York. *Beginning CSS: Cascading Style Sheets for Web Design*. Wrox Press, segunda edition, 2007.
- [20] C. Ullman and L. Dykes. *Beginning Ajax*. Wiley Publishing Inc., 2007.
- [21] Web Hypertext Application Technology Working Group (WHATWG). *Html 5* (2008, diciembre). disponible en: <http://whatwg.org/html5>.
- [22] Internet Engineering Task Force (IETF). *Xml media types* (2008, diciembre). disponible en: www.ietf.org/rfc/rfc3023.txt.
- [23] W3C. *Extensible markup language (xml) 1.0 (fifth edition)* (2008, diciembre). disponible en: <http://www.w3.org/TR/REC-xml/>.
- [24] W3C. *Namespaces in xml 1.0 (second edition)* (2008, diciembre). disponible en: <http://www.w3.org/TR/REC-xml-names/>.
- [25] W3C. *Xsl transformation (xslt) version 1.0* (2008, diciembre). disponible en: <http://www.w3.org/TR/xslt>.
- [26] W3C. *Document object model (dom) level 1 specification version 1.0* (2008, diciembre). disponible en: <http://www.w3.org/TR/REC-DOM-Level-1/>.
- [27] W3C. *Document object model (dom) level 2 core specification version 1.0* (2008, diciembre). disponible en: <http://www.w3.org/TR/DOM-Level-2-Core/>.
- [28] W3C. *Synchronized multimedia* (2008, diciembre). disponible en: <http://www.w3.org/AudioVideo/>.
- [29] L. Shklar and R. Rosen. *Web Application Architecture: Principles, Protocols and Practices*. John Wiley and Sons Inc., 2003.

- [30] The Apache Software Foundation. Apache tomcat (2008, diciembre). disponible en: <http://tomcat.apache.org/>.
- [31] Adobe. Jrun 4 (2008, diciembre). disponible en: <http://www.adobe.com/products/jrun/>.
- [32] ORACLE. Oracle weblogic server 10g standard edition (2008, diciembre). disponible en: <http://www.oracle.com/appserver/weblogic/standard-edition.html>.
- [33] M. Rydin G. Shachor, A. Chace. *JSP tag libraries*. Manning Publications Co., Greenwich, 2001.
- [34] Michael DeMers. *Fundamentals of geographic information systems*. John Wiley and Sons Inc., 3rd edition, 2005.
- [35] Inc. Open GIS Consortium. Opengis simple features specification for sql (2008, diciembre). disponible en: <http://www.opengeospatial.org/standards>.
- [36] A. K. W. Yeung and G. B. Hall. *Spatial Database Systems. Design, implementation and project management*, volume 87. Springer, 2007.
- [37] J.D. Taylor B.H. West, E.H. Griessback and L.T. Taylor. *The Prentice Hall Encyclopedia of Mathematics*. Prentice Hall, 1982.
- [38] J.F. Corbett. Topological principles in cartography. Technical Report 48, U.S. Department of Commerce, Bureau of the census, Washington, 1979.
- [39] ESRI. Arcview (2008, diciembre). disponible en: <http://www.esri.com/software/arcview/>.
- [40] ESRI. Esri (2008, diciembre). disponible en: <http://www.esri.com>.
- [41] D. Strand. Shapefiles shape gis data transfer standards. *GIS World*, 11(5):28–29, 1998.
- [42] R.H. Güting. An introduction to spatial databases. *VLDB Journal*, (4):357–399, 1994.
- [43] P. Rob and C. Coronel. *Database Systems: Design, Implementation and Management*. Thomson Learning, 5th edition, 2002.
- [44] ORACLE. Oracle spatial (2008, diciembre). disponible en: <http://www.oracle.com/technology/products/spatial/index.html>.
- [45] IBM. Db2 spatial extender (2008, diciembre). disponible en: <http://www-01.ibm.com/software/data/spatial/db2spatial/>.
- [46] ESRI. Arcgis-geodatabase (2008, diciembre). disponible en: <http://www.esri.com/software/arcgis/geodatabase/index.html>.

- [47] Refrations Research. Postgis (2008, diciembre). disponible en: <http://postgis.refrations.net/>.
- [48] PostgreSQL Global Development Group. Postgresql (2008, diciembre). disponible en: <http://www.postgresql.org/>.
- [49] IBM. Informix spatial blade (2008, diciembre). disponible en: <http://www-01.ibm.com/software/data/informix/blades/spatial/>.
- [50] ORACLE. Oracle enterprise manager (2008, diciembre). disponible en: <http://www.oracle.com/enterprisemanager/>.
- [51] Microsoft. Sql server 2008 (2008, diciembre). disponible en: <http://www.microsoft.com/sqlserver/2008>.
- [52] PostLBS. Proyecto pgrouting (2008, diciembre). disponible en: <http://pgrouting.postlbs.org/>.
- [53] University of Minnesota. Umn mapserver (2008, diciembre). disponible en: <http://mapserver.org/>.
- [54] OASIS. Modelo de referencia para una arquitectura orientada a servicios (2008, diciembre). disponible en: <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>.
- [55] Inc. Open GIS Consortium. Geography markup language (2008, diciembre). disponible en: <http://www.opengeospatial.org/standards/gml>.
- [56] Jun Zhang, Jihong Guan, Jianhua Zhang, and Junpeng Chen. Geographic information integration and publishing based on gml and svg. *Computer and Information Technology, 2004. CIT '04. The Fourth International Conference on*, pages 764–769, 14-16 Sept. 2004.
- [57] C. Li and D. Chenggang. A study on the web-gis-based information system with a particular reference to huludao city. *Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on*, pages 2888–2890, July 31 2006-Aug. 4 2006.
- [58] Ian Sommerville. *Ingeniería de Software*. Addison-Wesley, 2006.
- [59] C. Szyperski. *Component Software: Beyond Object-oriented Programming*. Addison-Wesley, 1998.
- [60] Prototype (2008, diciembre). Disponible en: <http://www.prototypejs.org/>.
- [61] Scriptaculous (2008, diciembre). Disponible en: <http://script.aculo.us/>.
- [62] Sun Microsystems. Java server pages standard tag library (2008, diciembre). disponible en: <http://java.sun.com/products/jsp/jstl/>.

-
- [63] Struts (2008, diciembre). Disponible en: <http://struts.apache.org/>.
- [64] Java Server Faces (2008, diciembre). Disponible en: <http://java.sun.com/javaee/jaserverfaces/>.
- [65] Spring (2008, diciembre). Disponible en: <http://www.springframework.org/>.
- [66] Tapestry (2008, diciembre). Disponible en: <http://tapestry.apache.org/>.
- [67] Google Chart (2008, diciembre). Disponible en: <http://code.google.com/apis/chart/>.
- [68] Microsoft. Towards 2020 science: A draft roadmap, disponible en: http://research.microsoft.com/en-us/um/cambridge/projects/towards2020science/downloads/t2020s_roadmap.pdf.
- [69] San Murugesan. Understanding web 2.0. *IEEE IT*, pages 34–41, Julio-Agosto, 2007.